2012

A Hierarchical Clustering and Validity Index for Mixed Data

Rui Yang
Iowa State University

A hierarchical clustering and validity index for mixed data

by

Rui Yang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Sigurdur Olafsson, Major Professor
Dianne Cook
Heike Hofmann
John Jackman
Jo Min

Iowa State University

Ames, Iowa

2012

ACKNOWLEDGEMENTS

I would like to thank everyone who support and encourage me while I completed my degree. Looking back, I could not have asked for a better person to be my major advisor than Dr. Sigurdur Olafsson. He allowed me the space to think creatively, but he was always there when I needed help or guidance.

I would also like to thank my committee members, Dr. Jo Min, Dr. John Jackman, Dr. Dianne Cook, and Dr. Heike Hofmann, for their concern, advice and encouragement for improving the quality of the thesis presented in every possible way.

Most importantly, I thank my husband, Jian Fan, for providing unwavering support for my pursuit of this dream, and my best friend, Hui Bian, who has always been my biggest emotional support.

Finally, I would like to express my greatest appreciation to my family and friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This study develops novel approaches to partition mixed data into natural groups, that is, clustering datasets containing both numeric and nominal attributes. Such data arises in many diverse applications. Our approach addresses two important issues regarding clustering mixed datasets. One is how to find the optimal number of clusters which is important because this is unknown in many applications. The other is how to group the objects "naturally" according to a suitable similarity measurement. These problems are especially difficult for the mixed datasets since they involve determining how to unify the two different representation schemes for numeric and nominal data.

To address the issue of constructing clusters, that is, to naturally group objects, we compare the performance of four distances capable of dealing with the mixed datasets when incorporating into a classical agglomerative hierarchical clustering approach. Based on these results, we conclude that the so-called co-occurrence distance to measure the dissimilarity performs well as this distance is found to obtain good clustering results with reasonable computation, thus balancing effectiveness and efficiency.

The second important contribution of this research is to define an entropy-based validity index to validate the sequence of partitions generated by the hierarchical clustering with the co-occurrence distance. A cluster validity index called the *BK* index is modified for mixed data and used in conjunction with the proposed clustering algorithm. This index is compared to three well-known indices, namely, the Calinski-Harabasz index (*CH*), the Dunn index (*DU*), and the Silhouette index (*SI*). The results show that the modified *BK* index outperforms the three other indices for its ability to identify the true number of clusters.

Finally, the study also identifies the limitation of the hierarchical clustering with a co-occurrence distance, and provides some remedies to improve not only the clustering accuracy but especially the ability to correctly identify best number of classes of the mixed datasets.

CHAPTER 1 INTRODUCTION

1.1 Motivation

Clustering is one of the fundamental techniques in data mining. The primary objective of clustering is to partition a set of objects into homogeneous groups (Jain and Dubes, 1988). An effective clustering algorithm needs a suitable measure of similarity or dissimilarity, so a partition structure would be identified in the form of "natural groups", where objects that are similar tend to fall into the same group and objects that relatively distinct tend to separate into different groups. Clustering has been extensively applied in diverse fields, including healthcare systems (Mateo et al., 2008), customer relations management (Jing et al., 2007), manufacturing systems (Suikki et al., 2006), biotechnology (Kim et al., 2009), finance (Liao et al., 2008), and geographical information systems (Touray et al., 2010).

Many algorithms that form clusters in numeric domains have been proposed. The majority exploit inherent geometry or density. This includes classical *k*-means (Kaufman and Rousseew, 1990; Jing et al., 2007) and agglomerative hierarchical clustering (Day and Edelsbrunner, 1984; Yasunori et al., 2007). More recently several studies have tackled the problem of clustering and extracting from categorical data, i.e., batch self-organizing maps (Chen and Marques, 2005), matrix partitioning method (Jiau et al., 2006), *k*-distributions (Cai et al., 2007), and fuzzy *c*-means (Brouwer and Groenwold, 2010). However, while the majority of the useful data is described by a combination of mixed features (Li and Biswas, 2002), traditional clustering algorithms are designed primarily for one data type. The literature on clustering mixed data is still relatively sparse (Hsu et al., 2007; Ahmad and Dey, 2007; Lee and Pedrycz, 2009) and more work is needed in this area.

The main obstacle to clustering mixed data is determining how to unify the distance representation schemes for numeric and categorical data. Numeric clustering adopts distance metrics while symbolic clustering uses a counting scheme to calculate conditional probability estimates as a means for defining the relation between groups. The pragmatic methods that convert one type of attributes to the other and then apply traditional single-type clustering algorithms may lead to significant loss of information. If categorical data with a large domain

is converted to numeric data by binary encoding, more space and time are introduced. Moreover, if quantitative and binary attributes are included in the same index, these procedures will generally give the latter excessive weight (Goodall, 1966).

Apart from the need for a suitable distance measure for mixed data, another critical issue is how to evaluate clustering structures objectively and quantitatively, that is, without using the domain knowledge and expert experience. The need to estimate the number of clusters in continuous data has led to the development of a large number of what is usually called validity criteria (Halkidi et al., 2002; Kim and Ramakrishna, 2005), but there are few criteria for evaluating partitions produced from categorical clustering (Celeux and Govaert, 1991; Chen and Liu, 2009). To the best of our knowledge there is no literature that satisfactorily addresses the cluster validation problems related to data with both discrete and real features.

The limitations of existing clustering methodologies and criterion functions in dealing with mixed data motivate us to develop clustering algorithms that can better handle both numeric and categorical attributes.

1.2 Objective

As stated above, this dissertation addresses the question of how to partition mixed data into natural groups efficiently and effectively. Later, the proposed approach will be applied to identify the "optimal" classification scheme among those partitions. The extension of clustering to a more general setting requires significant changes in algorithm techniques in several fundamental respects. To tackle the objectives stated above, the following three research tasks will therefore be addressed:

(1) We will develop an agglomerative hierarchical clustering method for clustering mixed datasets and investigate the performance of various distance measurements that represent both data types.

As mentioned above, the traditional way of converting data into a single type has many disadvantages. Within the context of an agglomerative hierarchical clustering method, we will investigate quantitative measures of similarity among objects that could keep not only the structure of categorical attributes but also relative distance of numeric values. Specifically, the measurements will be the co-occurrence distance (Ahmad and Dey, 2007),

the *k*-prototype distance (Huang, 1998), the optimal weight distance (Modha and Spangler, 2003), and the Goodall distance (Goodall, 1966). In the literature, the first three distances have been applied in *k*-means families, and the last is used in an agglomerative hierarchical clustering called the SBAC method (Li and Biswas, 2002). Our goal is to choose the distance measure that not only produces a low error rate in partitioning but is also suitable to search for the proper number of clusters.

(2) **We will investigate how to determine the optimal number of classes in mixed datasets.**

Evaluation of clustering structures can provide crucial insights about whether the clustering partition derived is meaningful or not. For numeric clustering, the number of clusters can be validated through geometry shape or density distribution, while cluster entropy and categorical utility are frequently used for categorical clustering. We will investigate how to extend the extant validity indices and make them capable of handling both data types. Specifically, we will investigate two approaches since they can integrate current validation methods smoothly. First, if a quantitative distance would represent numeric and categorical dissimilarities in a compatible way, then this geometric-like distance may be exploited in traditional numeric validation methods like the Calinski-Harabasz index, in which the optimal number of clusters would be determined by minimizing the intra-cluster distance while maximizing the inter-cluster distance. Second, it is also possible to calculate the entropies of cluster structures over both components. A low entropy is desirable since it indicates an ordered structure. We claim that the increments of expected entropies of optimal clusters structures over a series of successive cluster numbers would indicate the optimal number of clusters.

(3) We will explore the property of the proposed algorithm.

There is no cure-all algorithm for clustering problems. It is important to understand which datasets would be much more applicable to be analyzed by the new algorithm. Some testing datasets with various characteristics will be generated to investigate specific properties. Certainly, these properties could guide some data preprocessing operations on real-world datasets such as a feature selection.

1.3 Overview

The outcome of the research is a framework that combines a hierarchical clustering integrated with a co-occurrence distance and an extension of the *BK* index to search for the best number of the classes. Figure 1 shows this procedure. As illustrated, it contains six main steps to recover the underlying structure of mixed datasets under the assumption that the number of classes is unknown.

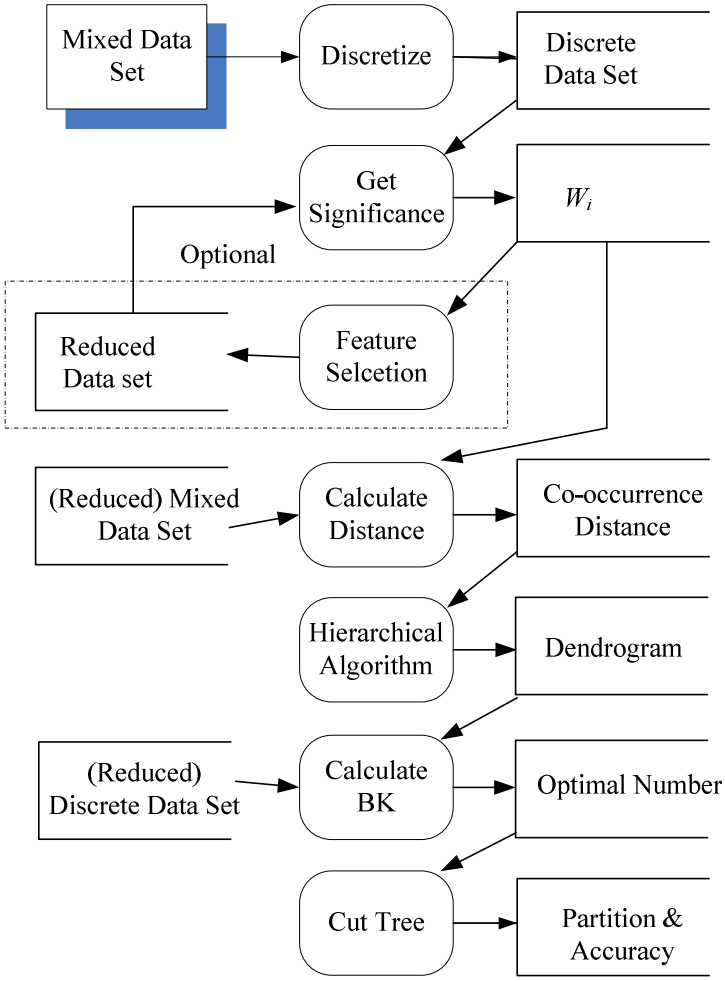

Figure 1: The proposed clustering framework.

The framework works as follows: (1) calculate the significance of each attribute, in which the numeric part will be used as weights in the co-occurrence distance; (2) conduct a feature selection according to the order of attributes' significance and the properties of the proposed algorithm. Thus, we can generate a reduced dataset that would be much more applicable to this algorithm and get better result. This step is optional, exploratory, and

iterative; (3) calculate the co-occurrence distance for the dataset of interest or the reduced mixed dataset; (4) construct a dendrogram by a hierarchical algorithm; (5) determine the optimal number of the classes by using *BK* index; (6) cut the tree according to this proper number and report the results.

1.4 Summary

All of the research tasks are either completely or partially new to the literature. The extended clustering applicable to arbitrary collections of datasets and the validity index for mixed datasets are particular novel and make significant contributions. The contributions in this study can be thus summarized as follows.

(1) Compare four distances capable of handling with the mixed dataset when used with hierarchical clustering.

(2) Identify limitations of the hierarchical clustering with a co-occurrence distance and propose solutions.

(3) Define a validity index to search the optimal number of clusters.

The remainder of the dissertation is organized as follows. In Chapter 2, we survey the related literature. In Chapter 3, we compare the four distances when used in an agglomerative clustering algorithm, and then choose the co-occurrence distance. The limitation of the proposed algorithm is identified. The corresponding solutions are provided. In Chapter 4, a validity index is integrated with the proposed algorithm to estimate the number of clusters for mixed data with numeric and categorical features. We conclude and suggest our future studies in Chapter 5.

CHAPTER 2 REVIEW OF LITERATURE

We will briefly review the literature on cluster analysis and cluster validation. The first section provides a basic understanding of clustering methods, specifically on categorical clustering and mixed clustering. Then we introduce several well-known validity indices to determine the optimal number of clusters.

2.1 Cluster Analysis

Cluster analysis was first proposed in numeric domains, where a distance is clearly defined. Then it extended to categorical data. However, much of the data in the real world contains a mixture of categorical and continuous features. As a result, the demand of cluster analysis on the mixed data is increasing.

2.1.1 Numeric Clustering

Clustering is to partition the data into groups where objects that are similar tend to fall into the same groups and objects that are relatively distinct tend to separate into different groups. Traditional clustering methodologies handle datasets with numeric attributes. The proximity measure can be defined by geometrical distance. A set of data with n objects (o_1, $\cdots$, o_n) are divided into k disjoint clusters (C_1, $\cdots$, C_k), called partition $P(k)$. n is the number of objects in the dataset and n_i the number of objects in the i_{th} cluster. $D(C_i, C_j)$ is the distance between the i_{th} cluster and the j_{th} cluster. $d(o_i, o_j)$ is the distance between the i_{th} object and the j_{th} object. The centroid of the i_{th} cluster is defined as $z_i = \frac{1}{n_i}\sum_{o \in C_i} o$. $Z = \{z_1, \cdots, z_k\}$ is a set of k center locations. $D(Z, o_i)$ is the shortest distance between object i and its nearest center.

Clustering constructs a flat (non-hierarchical) or hierarchical partitioning of the objects. Hierarchical algorithms use the distance matrix as input and create a sequence of nested partitions, either from singleton clusters to a cluster including all individuals or vice versa. Some details on agglomerative methods are provided here since the divisive clustering is not commonly used in practice. To begin, the n objects form n singleton clusters. The clusters with the minimal distance are merged. The distances between the new generated

cluster and others will be updated according to some linkage method. The searching for two clusters with the minimal distance and the merging process continue until all objects in the same cluster. The commonly used linkage methods are listed as follows, along with the definitions of inter-cluster distances and update rules.

(1) The single linkage defines the cluster distance as the smallest distance of a pair of objects in two different clusters. It is known as the nearest neighborhood method, which tends to cause chaining effect.

$$D(C_i, C_j) = \min_{o_i \in C_i, o_j \in C_j} d(o_i, o_j) \tag{2.1}$$

$$D(C_k, (C_i, C_j)) = \min(D(C_k, C_i), (C_k, C_j)) \tag{2.2}$$

(2) The complete linkage picks the furthest objects in two different clusters as the cluster distance.

$$D(C_i, C_j) = \max_{o_i \in C_i, o_j \in C_j} d(o_i, o_j) \tag{2.3}$$

$$D(C_k, (C_i, C_j)) = \max(D(C_k, C_i), (C_k, C_j)) \tag{2.4}$$

(3) The average linkage uses the average distance between all pairs of objects in cluster C_i and cluster C_j.

$$D(C_i, C_j) = \frac{1}{n_i n_j} \sum_{o_i \in C_i, o_j \in C_j} d(o_i, o_j) \tag{2.5}$$

$$D(C_k, (C_i, C_j)) = \frac{n_i}{n_i + n_j} D(C_k, C_i) + \frac{n_j}{n_i + n_j} D(C_k, C_j) \tag{2.6}$$

(4) The centroid linkage takes the distance between the centroids of two clusters as the cluster distance.

$$D(C_i, C_j) = d(z_i, z_j) \tag{2.7}$$

$$D(C_k, (C_i, C_j)) = \frac{n_i}{n_i + n_j} D(C_k, C_i) + \frac{n_j}{n_i + n_j} D(C_k, C_j) - \frac{n_i n_j}{(n_i + n_j)^2} D(C_i, C_j) \tag{2.8}$$

(5) The Ward's linkage is called the minimum variance method since it uses the increment of the within-class sum of squared errors when joining clusters C_i and C_j as the cluster distance between cluster C_i and cluster C_j.

$$D(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} d^2(z_i, z_j) \tag{2.9}$$

$$D(C_k, (C_i, C_j)) = \frac{n_i + n_k}{n_i + n_j + n_k} D(C_k, C_i) + \frac{n_j + n_k}{n_i + n_j + n_k} D(C_k, C_j) - \frac{n_k}{n_i + n_j + n_k} D(C_i, C_j) \tag{2.10}$$

Non-hierarchical partition clustering employs an iterative approach to group data into a pre-specified number by minimizing a sum of weighted within-cluster distances between every object and its cluster center.

$$\text{minimize } f(Z) = \sum_{i=1}^{n} w_i D(Z, o_i) \tag{2.11}$$

The weight, $w_i > 0$, modifies the distances. If treat the weighted distance $w_i D(Z, o_i)$ as a "cost", we formulate it as a standard discrete optimization problem.

Let x_{ij} be a decision variable

$$x_{ij} = \begin{cases} 1 & \text{if object } i \text{ is assigned to the } j_{th} \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$$

To ensure that every object is assigned to exactly one cluster, it has

$$\sum_{j=1}^{k} x_{ij} = 1 \qquad \forall i$$

The interpretation of the notation is as

i : index of objects;

j : index of clusters;

d_{ij} : distance between object i and the center of cluster j.

$$\text{minimize } f(X) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_i d_{ij} x_{ij} \tag{2.12}$$

subject to

$$\sum_{j=1}^{k} x_{ij} = 1 \qquad \forall i$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j$$

By selecting a vector of cluster centers from the set of feasible alternatives defined by the constraints, the model achieves the minimum total cost, namely, the minimum total weighted within-group distance over all groups.

Unfortunately, this problem is NP-hard even for $k = 2$ (Drineas et.al, 2004). It is impossible to find exact solutions in polynomial time unless P = NP. However, there are some efficient approximate approaches, such as *k*-means algorithms.

2.1.2 Categorical Clustering

For categorical data which has no order relationships, conceptual clustering algorithms based on hierarchical clustering were proposed. These algorithms use conditional probability estimates to define relations between groups. Intra-class similarity is the probability $\Pr(a_i = v_{ij} | C_k)$ and inter-class dissimilarity is the probability $\Pr(C_k | a_i = v_{ij})$, where $a_i = v_{ij}$ is an attribute-value pair representing the i_{th} attribute takes its j_{th} possible value. *Category Utility* (*CU*) is a heuristic evaluation measure (Fisher, 1987) to guide construction of the tree in systems COBWEB (Huang and Ng, 1999) or its derivatives, e.g., COBWEB/3 (McKusick and Thomson, 1990), and ITERATE (Biswas et al., 1998). *CU* attempts to maximize both the probability that two objects in the same cluster have attribute values in common and the probability that objects from different clusters have different values. ROCK (Guha et al., 1999) is a clustering algorithm that works for both boolean and categorical attributes. This algorithm employs the concept of links to measure the similarity between a pair of data points. The number of links between a pair of points is the number of common neighbors shared by the points. Clusters are merged through hierarchical clustering which checks the links while merging clusters. The main objective of the algorithm is to group together objects that have more links. CACTUS (Ganti et al., 1999) is a hierarchical algorithm to group categorical data by looking at the *support* of two attribute values. *Support* is the frequency of two values appearing in objects together. The higher the support is, the more similar the two attribute values are. The two attribute values are strongly connected if their support exceeds the expected value with the assumption of attribute-independence. This concept is extended to a set of attributes that pair wise strongly connected. Finding the co-occurrence of a set of attribute values is intensive in computational complexity.

2.1.3 Mixed Clustering

Clustering algorithms are designed for either categorical data or numeric data. However, in the real world, a majority of datasets are described by a combination of continuous and categorical features. A general method is to transform one data type to another. In most cases, nominal attributes are encoded by *simple matching* or *binary mapping*, and then clustering is performed on the new-computed numeric proximity.

Binary encoding transforms each categorical attribute to a set of binary attributes, and then encodes a categorical value to this set of binary values. *Simple matching* generates distance measurement in such a way that yields a difference of zero when comparing two identical categorical values, and a difference of one while comparing two distinct values. However, the coding methods have the disadvantages of (1) losing information derivable from the ordering of different values, (2) losing the structure of categorical value with different levels of similarity, (3) requiring more space and time when the domain of the categorical attribute is large, (4) ignoring the context of a pair of values, e.g., the co-occurrence with other attributes, and (5) giving different weight to the attributes according to the number of different values they may take. Moreover, if quantitative and binary attributes are included in the same index, these procedures will generally give the latter excessive weight (Goodall, 1966).

An alternative approach is to discretize numeric values and then apply symbolic clustering algorithms. The discretization process often loses the important information especially the relative difference of two values for numeric features. In addition, it causes boundary problem when two close values near the discretization boundary may be assigned to two different ranges. Another difficult problem is to estimate the optimal intervals during discretization.

Huang (1998) extended *k*-modes to mixed datasets and developed *k*-prototype algorithm. The distances of two types of features are separately calculated. The numerical distances are measured by Euclidean distances, while the categorical distances are measured by simple matching. The centers of categorical attributes are defined as the modes in the cluster. Ahmad and Dey (2007) proposed a fuzzy prototype *k*-means algorithm. Similar to *k*-prototype, the cost function is made up of two components. The difference is that the

categorical distances are measured by the co-occurrence of two attributes and the categorical cluster centers are the lists of values in every attribute with their frequencies in the cluster. Modha and Spangler (2003) used *k*-means to cluster mixed datasets, but they carefully chose the weights for different features by minimizing the ratio of the between-cluster scatter matrix and the with-cluster scatter matrix of the distorted distance.

ECOWEB (Reich and Fenves, 1991) defines *Category Utility* measurement in numeric attributes by approximation of the probability in some user-described interval, which has greatly impact on the performance. AUTOCLASS (Cheesman and Stutz, 1995) assumes a classical finite mixture distribution model on the data and uses a Bayesian method to maximize the posterior probability of the clustering partition model given the data. The number of classes in the data is pre-specific. The computational complexity is extremely expensive. SBAC (Li and Biswas, 2002) is a hierarchical clustering of mixed data based on Goodall similarity measurement with the assumption of attribute-independence. The distance exploits the property that a pair of the objects is closer than other pairs if it has an uncommon feature. This algorithm is computationally prohibitive and demands huge memory. Hsu and his colleagues (2007) exploited the semantics property in the domain of categorical attributes and represented each attribute with a tree structure whose leaves are the possible values of this attribute and the links associate with some user-specified weights. This hierarchical distance scheme is integrated with agglomerative hierarchical clustering and compared to binary coding and simple matching.

Some fuzzy clustering algorithms are proposed recently to attack the dataset with mixed features. Unlike hard clustering where each object belongs to only one cluster, fuzzy clustering algorithms assign each object to all of the clusters with a certain degree of membership. Yang et al. (2004) investigated symbolic dissimilarity that is originally proposed by Gowda and Diday (1991) and modified the three components parts of dissimilarity measure. This fuzzy clustering has the strength to handle the categorical data and fuzzy data. GFCM (Lee and Pedrycz, 2009) used a fuzzy center instead of a singleton prototype for the categorical components, which took a list of partial values of a categorical attribute with their frequencies in the cluster. The size of the values in the prototype is an input parameter. Besides searching for optimal membership matrix and prototype matrix, the

algorithm has to choose a set of values from a categorical attribute domain and present them in the prototype. The size of the labels and the fuzzification coefficients affect the performance.

2.2 Cluster Validation

Clustering algorithms expose the inherent partitions in the underlying data, while cluster validation methods are able to evaluate the result clusters quantitatively and objectively, e.g., whether the cluster structure is meaningful or just an artifact of the clustering algorithm. There are two main categories of testing criteria, known as external indices and internal indices. External indices are distinguished from internal indices by the present of priori information of known categories.

2.2.1 External Indices

Given a priori known cluster structure (P) of the data, external indices evaluate a clustering structure resulting from cluster algorithms (P') based on counting the pairs of points on which two partitions agreement and disagreement. A pair of points can fall into one of the four cases as below:

a : number of point pairs in the same cluster in both P and P'

b : number of point pairs in the same cluster in P but not in P'

c : number of point pairs in the same cluster in P' but not in P

d : number of point pairs in different clusters under both P and P'

Wallace (1983) proposed the two asymmetric criteria W_1, W_2 as

$$W_1(P, P') = \frac{a}{a+b} \quad \text{and} \tag{2.13}$$

$$W_2(P, P') = \frac{a}{a+c} \tag{2.14}$$

representing the probability that a pair of points which are in the same cluster in P (respectively, P') are also in the same cluster under the other clustering.

Fowlkes and Mallows (1983) took the geometric mean of the asymmetric Wallace indices and introduced a symmetric criterion

$$FM(P, P') = \sqrt{\frac{a}{a+c}\frac{a}{a+b}}. \tag{2.15}$$

The Fowlkes-Mallows index assumes the two partitions are independent.

The Rand index emphasizes the probability that a pair of points in the same group or in different groups in both partitions while Jaccard's coefficient does not take an account into 'conjoint absence' and only measures the portion of pairs in the same cluster.

The Rand index (Rand, 1971)

$$R(P, P') = \frac{a+d}{a+b+c+d} \tag{2.16}$$

The Jaccard index (Jain and Dubes, 1988)

$$J(P, P') = \frac{a}{a+b+c} \tag{2.17}$$

2.2.2 Internal Indices

Internal indices are validation measures which evaluate clustering results using only information intrinsic to the underlying data. Without true cluster labels, estimating the number of clusters, k, in a given dataset is a central task in cluster validation. Overestimation of k complicates the true clustering structure, and makes it difficult to interpret and analyze the results; on the other hand, underestimation causes the loss of information and misleads the final decision. In the following section, we will briefly review several well-known indices.

One of the oldest and most cited indices is proposed by Dunn (Dunn, 1974) to identify the clusters that are compact and well separated by maximizing the inter-cluster distance while minimizing the intra-cluster distance. The Dunn index for k clusters is defined as

$$k^* = \arg\max_{k \geq 2} \left\{ DU(k) = \min_{i=1,\cdots,k} \left(\min_{j=i+1,\cdots,k} \left(\frac{D(C_i, C_j)}{\max_{m=1,\cdots,k} diam(C_m)} \right) \right) \right\}, \tag{2.18}$$

where $D(C_i, C_j)$ is the distance between two clusters C_i and C_j as the minimum distance between a pair of objects in the two different clusters separately and the diameter of cluster C_m, $diam(C_m)$, as the maximum distance between two objects in the cluster. The optimal

number of clusters is calculated at the largest value of the Dunn index. The Dunn index is sensitive to noise. By redefinitions of the cluster diameter and the cluster distance, a family of cluster validation indices is proposed (Bezdek and Pal, 1998).

Based on the ratio of the between-cluster scatter matrix ($\mathbf{S_B}$) and the within-cluster scatter matrix ($\mathbf{S_W}$), the Calinski-Harabasz index (Calinski and Harabasz, 1974) is the best among the top 30 indices ranked by Milligan and Cooper (1985). The optimal number of clusters is determined by maximizing $CH(k)$.

$$k^* = \arg\max_{k\geq 2}\left(CH\left(k\right) = \frac{Tr(\mathbf{S_B})/(k-1)}{Tr(\mathbf{S_W})/(n-k)} \right). \tag{2.19}$$

Similar to the Calinski-Harabasz index, the Davies-Bouldin index (Davies and Bouldin, 1979) obtains clusters with the minimum intra-cluster distance as well as the maximum distance between cluster centroids. The minimum value of the index indicates a suitable partition for the dataset.

$$k^* = \arg\min_{k\geq 2}\left\{ DB(k) = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1,\cdots,k,i\neq j}\left(\frac{diam(C_i)+diam(C_j)}{d(z_i,z_j)} \right) \right\} \tag{2.20}$$

where the diameter of a cluster is defined as

$$diam(C_i) = \sqrt{\frac{1}{n_i}\sum_{o\in C_i} d(o,z_i)^2}. \tag{2.21}$$

The Silhouette index (Kaufman and Rousseeuw, 1990) computes for each object a width depending on its membership in any cluster. For the i_{th} object, let a_i be the average distance to other objects in its cluster and b_i the minimum of the average dissimilarities between object i and other objects in other clusters. The silhouette width is defined as $(b_i - a_i)/\max\{a_i, b_i\}$. Silhouette index is the average Silhouette width of all the data points. The partition with the highest $SI(k)$ is taken to be optimal.

$$k^* = \arg\max_{k\geq 2}\left(SI\left(k\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{(b_i-a_i)}{\max(b_i,a_i)} \right) \tag{2.22}$$

The Geometric index (Lam and Yan, 2005) is recently proposed to accommodate data with clusters of different densities and overlap clusters. The optimal number of clusters is found by minimizing the $GE(k)$ index. Let d be the dimensionality of the data and λ_{pq} the

eigenvalue of the covariance matrix from the data. $D(C_i, C_j)$ is the inter-cluster distance between cluster i and cluster j. The GE index is constructed as

$$k^* = \arg\min_{k\geq 2}\left\{ GE(k) = \max_{1\leq i\leq k}\left(\frac{\left(2\sum_{j=1}^{d}\sqrt{\lambda_{ji}}\right)^2}{\min_{1\leq j\leq k, i\neq j} D(C_i, C_j)} \right)\right\}. \tag{2.23}$$

Unlike the criteria mentioned above, which employ the geometric-like distance, CU and entropy-based methods use the counting scheme to evaluate the performance of a categorical clustering algorithm. CU of a partition with k clusters is defined in Eq. 2.24. A cluster solution with high CU is desired since it improves the likelihood of similar patterns falling into the same cluster.

$$k^* = \arg\max_{k\geq 2}\left\{ CU(k) = \sum_{l=1}^{k}\left\{ \frac{n_l}{n}\sum_i\sum_j\left[\Pr(a_i = v_{ij} \mid C_l)^2 - \Pr(a_i = v_{ij})^2 \right]\right\}\right\} \tag{2.24}$$

Entropy-based method computes the expected entropy of a partition with respect to a class attribute a_i. The smaller the expected entropy, the better quality of the partition with respect to a_i. It is expected that the expected entropy decreases monotonically as the number of clusters increases, but from some point onwards the decrease flattens remarkably. Rather than searching for the location of an "elbow" on the plot of the expected entropy versus the number of clusters, Chen and Liu (2009) calculated the second order difference of incremental expected entropy of the partition structure, which is called the BK index. The largest value indicates an elbow point which is the potential number of clusters.

$$k^* = \arg\min_{k\geq 2}\left\{ \mathrm{E}_{a_i}(k) = -\sum_l \frac{n_l}{n}\sum_j \Pr(a_i = v_{ij} \mid C_l)\log\Pr(a_i = v_{ij} \mid C_l)\right\} \tag{2.25}$$

The significance test on external variables is other commonly used method. It compares the partitions using variables not used in the generation of those clusters.

Although the validity index for mixed features is relatively sparse, there are a few to evaluate fuzzy clustering algorithms based on the fuzzy partition matrices and/or dissimilarity among objects and prototypes. For example, Lee (2009) proposed an index called $CPI(k)$ as

$$k^* = \arg\max_{k \geq 2} \left\{ CPI(k) = \frac{1}{k} \sum_{l=1}^{k} \frac{\sum_{i \neq j} u_{li} u_{lj} sim(o_i, o_j)}{\sum_{i \neq j} u_{li} u_{lj}} - \frac{1}{k(k-1)} \sum_{s=1;t=1;s \neq t}^{k} \frac{\sum_{i \neq j} u_{si} u_{tj} sim(o_i, o_j)}{\sum_{i \neq j} u_{si} u_{tj}} \right\}, \quad (2.26)$$

where u_{ij} is the membership of object o_i in cluster j, $0 \leq u_{ij} \leq 1$. $sim(o_i, o_j)$ is any similarity measure between object o_i and o_j.

2.3 Summary and Discussion

Real-life systems are overwhelmed with large mixed datasets that include numeric and nominal data. However, the majority of the clustering algorithms are designed for one data type. This study will propose a novel approach to partition mixed dataset, evaluate the resulting cluster solutions, and determine the optimal number of clusters.

CHAPTER 3 HIERARCHICAL CLUSTERING FOR MIXED DATA

Many partitioning algorithms require the number of classes, k, as a user-specified parameter. However, k is not always available in many applications. Hierarchical clustering does not need this priori information. This method creates a sequence of nested partitions. In order to form a set of clusters, a cutting point is determined by using some expert experiences to interpret each branch in the dendrogram, or by applying some validity indices to estimate where the best levels are. Our algorithm can be divided into two main procedures. First, a cluster tree is constructed by a hierarchical algorithm in a bottom up manner; and then a search procedure is followed to obtain the optimal number of classes.

In this chapter, we present the first procedure that generates a cluster tree by hierarchical clustering on the distances from a co-occurrence measure. We compare four distances measurements capable of handling mixed data when used with agglomerative hierarchical clustering, and provide a solution in which the co-occurrence distance would outperform other distances. The performance is tested on some standard real-life as well as synthetic datasets.

3.1 Motivation

A distance measure that has been previously found to perform well for the fuzzy prototype k-means algorithm (Ahmad and Dey, 2007) will be adopted to define the proximity between pairs of objects. Without the assumption about data distribution, it considers the strong co-occurrence probability of two attribute values in a certain class.

Intuitively, some attribute values are associated with different classes. For example, the color of a banana is yellow while the color of a strawberry is red. If a basket has only strawberries and bananas and, by a chance, we pick a yellow fruit, then it must be a banana. Likewise, if we know the type of the fruit in this basket, then the color can be decided. Yellow and red have strong correlations with bananas and strawberries, respectively. The color of the fruit can distinguish each type of the fruit. In this context, we can assume the distance between yellow and red is large with respect to type of fruit. However, if we harvest yellow lemons and red tomatoes, and put them into this basket, then it is difficult to tell the

types of the fruits from their colors. In this situation, in terms of type of fruit, we can say a small distance between yellow and red. Therefore, a distance would be defined based on the co-occurrence of colors with respect to fruit types. A strong occurrence relationship between the two levels of color and type of fruit results in a large distance and vice versa.

Based on the power of an attribute to separate data into homogenous segments, Ahmad and Dey (2007) defined the distance between categorical attribute values and calculated the weights for numeric attributes by exploiting this co-occurrence relation. The overall distance is a sum of categorical and the weighted numeric distances and applied in the *k*-means algorithm to cluster mixed datasets. The comparative study showed good performance. Ahmad and Dey's fuzzy prototype *k*-means method does not work in applications without a known number of groups. Unlike Ahmad and Dey's method, therefore, our study employs this distance in a hierarchical algorithm to derive a tree structure, which can generate a series of partitions with successive cluster numbers. These partitions will be evaluated by the validity index proposed in the following chapter. In this section, we compare the hierarchical algorithm with four distances capable of handling mixed data types in terms of clustering accuracy, and exploit the properties of the datasets would take the advantage of the co-occurrence distance when used with agglomerative hierarchical clustering.

3.2 Background

Traditional approaches of clustering datasets with mixed data types adopt distance representations by converting one type of attributes to another. One way is to transfer categorical data into numeric data by simple matching or binary coding. On the other hand, the continuous attributes are discretized into categorical data. As mentioned in the preceding chapters, these two ways are not effective in dealing with the particular mixed datasets. Some researchers have made efforts to balance numeric and nominal distances. Huang (1998) introduced a weight factor for categorical distance in his *k*-prototype algorithm. Modha and Spangler (2003) went further and found the optimal weight that minimizes the within-cluster weighted distance while maximizing the between-cluster weighted distance. The SBAC method (Li and Biswas, 2002) adopted the Goodall distance based on the concept that two

species are closer if they have rarer characteristics in common. From the view of probability, the Goodall distance unites categorical and numeric distances within a common framework. Ahmad and Dey (2007) defines a distance by exploiting a co-occurrence relation of values in different attributes. The *k*-prototype, the optimal weight, the Goodall, and the co-occurrence distances are discussed in greater details below.

3.2.1 Notation

DS(*U*, *A*) represents a set of objects in terms of their attributes, where *U* is a nonempty finite set of objects and *A* is a nonempty finite set of attributes. For example, a strawberry and a banana are two objects in *U*, while color, type, and weight of the fruit are the attributes in *A*. Let *n* be the number of objects and *m* the number of attributes. There exists a function between the set of objects and each attribute such that $a_p : U \to V_{a_p}$ for any $a_p \in A$ (p = 1, 2, …, m), i.e., $a_p(x_i) \in V_{a_p}$ (i = 1, 2, …, n), $x_i \in U$, where V_{a_p} is called the domain of attribute a_p. $a_p(x_i)$ is the value of object x_i on attribute a_p. For example, the color of a banana is denoted as $a_{color}(banana) = yellow$. Generally speaking, the set of attributes can be divided into two subsets A_r and A_c according to data type, where A_r is the set of numeric attributes and A_c the set of categorical attributes. Thus, $A = A_r \bigcup A_c$. If *A* is the set including color, type, and weight of the fruit, then A_c is the set with color and type of the fruit while A_r contains the weight of the fruit. m_r and m_c are the numbers of numeric and categorical attributes, respectively. $m = m_r + m_c$. Given $a_p \in A$, $x, y \in U$, if $a_p(x) = a_p(y)$, then *x* and *y* are said to have no difference w.r.t. a_p. The distance of *x* and *y* on a_p is zero, denoted by $D^p(x, y) = 0$ when $a_p(x) = a_p(y)$. For example, if one fruit in the basket is a banana and another is a lemon, we know their colors are yellow. It can be denoted as $a_{color}(banana) = a_{color}(lemon)$, or $D^{color}(banana, lemon) = 0$. The total distance between two objects *x*, *y* is defined as $d(x, y)$.

Let $R_j = \{(x, y) \in U \times U : a_j(x) = a_j(y)\}$. Thus, the relation R_j partitions *U* into disjoint subsets according to values on attribute a_j. These subsets are called equivalence classes of a_j. The equivalence class including *x* is $S_j(x)$, $S_j(x) = \{ y \in U : (x, y) \in R_j \}$. For example, if *x* represents a banana, then $S_{color}(banana)$ contains all the fruits in the basket that have the

same color as the banana. Thus, $S_{color}(banana)$ is the set of all bananas and lemons in the basket. $W_j(B_j) = \{ S_j(x) : a_j(x) \in B_j, B_j \subset V_{a_p} \}$, the set of objects whose values on a_j are among B_j. If the objects having the same value w.r.t. a_j should all be included in one segment, either $W_j(B_j)$ or $U/W_j(B_j)$. $U/W_j(B_j)$ is called *the simply complement* of $W_j(B_j)$, usually denoted by $\sim W_j(B_j)$. If $W_{color}(\{\text{yellow}\})$ represents the set of fruits with a yellow appearance, then $\sim W_{color}(\{yellow\})$ is the set of fruits containing colors except yellow.

3.2.2 *k*-prototype Distance

In order to cluster mixed datasets, Huang (1998) used a user-specified weight γ to balance the distance over numeric and nominal attributes and applied this measure in his *k*-prototype algorithm. The numeric distance is the squared-Euclidean distance; and the categorical distance is simple matching. All numeric attributes are normalized to the range of [0, 1]. A small γ value indicates that the clustering is dominated by numeric attributes while a large γ value implies that categorical attributes dominate the clustering. Huang suggested the weight should be in the range of [0.5, 1.4]. The total distance between a pair of objects x and y is formulated as,

$$d(x,y) = \sum_{i \in A_r} \left\{ \frac{a_i(x) - a_i(y)}{\max(V_{a_i}) - \min(V_{a_i})} \right\}^2 + \gamma \sum_{i \in A_c} D^i(x,y), \tag{3.1}$$

and the computational complexity is $O(mn^2)$.

3.2.3 Optimal Weight Distance

Modha and Spangler (2003) used optimization techniques to further balance numeric and nominal distances. Instead of a user-specified weight, their method searches for an optimal weight to minimize the ratio of the within- and between-cluster weighted distances when the number of clusters is given.

The numeric features are standardized based on mean and standard deviance, and then the distance is found by taking the squared-Euclidean distance. Each categorical value is represented by a binary vector using 1-of-v encoding (v is the number of attribute values), and the distance is found by taking cosine distance. The optimal weight distance combines the weighted distances of the two data types.

$$d(x,y) = (1-w)\sum_{i\in A_r} \frac{(a_i(x)-a_i(y))^2}{\mathrm{var}(V_{a_i})} + w\sum_{i\in A_c} D^i(x,y) \tag{3.2}$$

In order to get the optimal weight, the number of clusters should be known in advance. Since the objective function of this minimization problem is nonlinear, it is hard to pursue an optimal solution. Modha and Spangler calculated the objective value by taking a large number in the interval [0, 1] in order to search the best one. The computational complexity is $O(imn^2)$ when choosing i iterations to search the weight.

3.2.4 Goodall Distance

Goodall (1966) proposed a similarity index based on the agreement that a pair of objects having an uncommon value of an attribute is closer than other pairs only possessing a common value among them. For example, a salmon and a bass have scales but a dolphin and a salmon have vertebra. Since there are more animals having vertebra than those having scales, a salmon is closer to a bass than to a dolphin. The author made an assumption about independent attributes. Li and Biswas (2002) adopted this distance in the SBAC method. The distance for non-identical nominal values is one, as $D^i(x, y) = 1$, $a_i(x) \neq a_i(y)$, $a_i \in A_c$. For example, in the case of the fruit basket, the distance between yellow and red is formulated as $D^{color}(banana, strawberry) = 1$. However, $D^{color}(banana, lemon)$ is much more complex. For a pair of identical nominal values, the distance is the sum of the possibilities of picking an identical value pair whose value is equally or more similar to the pair in question, that is, having lower or equal frequency. The formulation is as follows when $s = a_i(x) = a_i(y)$, $a_i \in A_c$.

$$D^i(x,y) = \sum_{r\in V_{a_i}, freq(r)\le freq(s)} \frac{freq(r)(freq(r)-1)}{n(n-1)} \tag{3.3}$$

The distance between identical numeric values is calculated as their nominal component using equation (3.3).

To calculate distance of two different numeric values, divide the domain into successive segments by the unique values of a numeric feature and count the frequency in every interval first. Sum the possibilities in a smaller range or equal-width range (l, m) but with less or equal frequency. Given $s = a_i(x)$, $t = a_i(y)$, $a_i(x) \neq a_i(y)$, $a_i \in A_r$, the distance on a numeric attribute A_i is defined as follows.

$$D^i(x,y) = \sum_{r \in V_{a_i}} \frac{freq(r)(freq(r)-1)}{n(n-1)} + \sum_{\substack{|m-l|<|t-s| \vee \{|m-l|=|t-s|, \\ freq(|m-l|) \le freq(|t-s|)\}}} \frac{2\,freq(l)(freq(m)-1)}{n(n-1)} \tag{3.4}$$

When having calculated the distances for each attribute, we use χ^2 transformation to get corresponding chi-square values. The sum of these values is distributed as χ^2 with the degree of freedom two times of the number of attributes. The probability of this sum is the Goodall distance of a pair of objects.

It takes $O(n+v_i \log v_i)$ to compute the nominal distance for a categorical attribute with v_i levels. For m_c attributes, it needs m_c such calls. Therefore, the running time is $O(nm_c+mq_c \log q_c)$, where $q_c = \max\{\|V_{a_i}\|, a_i \in A_c\}$. Sorting the intervals of a numeric attribute with v_i unique values requires $O(v_i^2 \log v_i)$. Given l_j, the number of the intervals in j_{th} equal-range intervals, the same-range intervals are sorted by their frequencies in $O(\sum_{j=1}^{v_i^2} l_j \log l_j)$, which is upper bounded by $O(v_i^2 l \log l)$ and l is the maximum number over all l_j. Accordingly, the computational complexity of numeric distances is $O(nm_r+m_r q_r^2 \log q_r+m_r\, q_r^2 l \log l)$; and the total running time is $O(nm+mq_c \log q_c+mq_r^2 \log q_r+mq_r^2 l \log l)$, where q_r is the number of maximum number of unique values in the numeric attributes. Even in an ordinary dataset, the number of q_r is huge.

3.2.5 Co-occurrence Distance

Ahmad and Dey (2007) exploit the property that if there is a stronger connection of values on a_i (e.g., s and t) with different values on a_j, then s and t are more powerful to separate a dataset into pure segments w.r.t. a_j. The extreme case is when s and t associate with different values in a_j.

Given $W_j(B_j)$, let $P_i(W_j(B_j)|s)$ be the conditional probability w.r.t. $W_j(B_j)$ when the value of object x on attribute a_i is s, and $P_i(\sim W_j(B_j)|s)$ the conditional probability of set $U/W_j(B_j)$ when the value of object x on attribute a_i is s, where $s \in V_{a_i}$, $a_i \in A_c$. These two formulations are

$$\mathrm{P_i}\left(W_j(B_j)\,|\,s\right)=\frac{\text{the number of } y, y\in W_j(B_j)\wedge a_i(y)=s}{\text{the number of } x, x\in U\wedge a_i(x)=s} \tag{3.5}$$

$$\mathrm{P_i}\left(\sim W_j(B_j)\,|\,s\right)=\frac{\text{the number of } y, y\in U/W_j(B_j)\wedge a_i(y)=s}{\text{the number of } x, x\in U\wedge a_i(x)=s}. \tag{3.6}$$

The definition of the distance of two levels in categorical attribute a_i with respect to attribute a_j is

$$D^{ij}(s,t)=\max_{B_j}\left\{\mathrm{P}_i(W_j(B_j)\,|\,s)+\mathrm{P}_i(\sim W_j(B_j)\,|\,t)-1.0\right\}, \tag{3.7}$$

where $s, t\in V_{a_i}$, $s\neq t$, $a_i, a_j\in A_c$. Ahmad and Dey (2007) showed an optimal solution would be obtained in polynomial algorithm in terms of $\left\|V_{a_j}\right\|$.

The distance of two levels in categorical attribute a_i is the average of $D^{ij}(s, t)$ over all categorical attributes but a_i. Given $s = a_i(x)$, $t = a_i(y)$, $s \neq t$, $a_i \in A_c$,

$$D^i(x,y)=\frac{1}{m_c-1}\sum_{j\neq i, a_j\in A_c} D^{ij}(s,t), \text{ and} \tag{3.8a}$$

$$Dist^i(s,t)\equiv D^i(x,y)=\frac{1}{m_c-1}\sum_{j\neq i, a_j\in A_c} D^{ij}(s,t). \tag{3.8b}$$

The distance between two values on numerical attribute $a_i \in A_r$ is

$$D^i(x,y)=\frac{\left|a_i(x)-a_i(y)\right|}{\max(V_{a_i})-\min(V_{a_i})}, \tag{3.9}$$

which is equivalent to normalizing numerical attribute a_i first and then taking the absolute value of the difference.

The distance between every pair of objects w.r.t. attribute set A is defined as

$$d(x,y)=\sum_{i\in A_r}\left\{w_i\frac{a_i(x)-a_i(y)}{\max(V_{a_i})-\min(V_{a_i})}\right\}^2+\sum_{i\in A_c} D^i(x,y), \tag{3.10}$$

where w_i is the weight of real attribute a_i. The weight w_i is introduced to modify numerical distances based on separating power to divide the dataset into pure segments. First, a numerical attribute $a_i \in A_r$ will be discretized into v intervals, where v is a predefined integer. w_i is calculated as Eq. 3.11 to reveal the significance of attribute a_i to separate the dataset. In

$$w_i = \frac{\sum_{(s,t)} Dist^i(s,t)}{v_i(v_i - 1)}, \tag{3.11}$$

s and t are the new assigned categorical values for discretized numerical attribute a_i and v_i is the number of categorical values, $v_i = \left\| V_{a_i} \right\|$.

The running time of calculating the co-occurrence distance is $O(n^2m+nm^2+q^3m^2)$, where $q = \max\{\left\| V_{a_i} \right\|, a_i \in A_c\}$.

3.3 Proposed Hierarchical Clustering Method

In this section, we will integrate an agglomerative clustering algorithm with the four distances measures capable of handling mixed datasets.

3.3.1 Agglomerative Hierarchical Algorithm

Agglomerative hierarchical algorithms start from singleton clusters and merge those clusters with minimal distances until all objects are included in one cluster. The distance between individual objects is as important in creating clusters as the cluster distance, but the cluster distance has greater weight on creating the final partition. Although there are a large number of distance definitions between a cluster and a newly formed cluster, we choose Ward's method to minimize the increase of the within-class sum of the squared errors since we wish the formed clusters would be compact, not chain-like or with one object. The within-class sum of the squared errors is the sum of squared-Euclidean distance between each object to its nearest cluster center and is formulated as

$$E = \sum_k \sum_i \left\| o_i - z_k \right\|^2 \tag{3.12}$$

where o_i is an object in the k_{th} cluster and z_k the centroid of this cluster. If two clusters C_i and C_j are merged, the increment of E will be calculated with the following equation:

$$\Delta E_{ij} = \frac{n_i n_j}{n_i + n_j} \left\| z_i - z_j \right\|^2. \tag{3.13}$$

Thus, the distance between two clusters C_i and C_j is defined as

$$D(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} d^2(z_i, z_j). \quad (3.14)$$

The distance between a cluster C_k and a newly emerged cluster (C_i and C_j) is determined by

$$D(C_k, (C_i, C_j)) = \frac{n_i + n_k}{n_i + n_j + n_k} D(C_k, C_i) + \frac{n_j + n_k}{n_i + n_j + n_k} D(C_k, C_j) - \frac{n_k}{n_i + n_j + n_k} D(C_i, C_j), \quad (3.15)$$

which would be used to update cluster distances after a merge.

The following procedure summarizes the agglomerative hierarchical algorithm.

INPUT. Distance matrix for each pair of objects $(x, y) \in (U \times U)$.

OUTPUT. A dendrogram.

Initial: Every object forms a singleton cluster. There are n clusters.

Step 1: Search the minimal cluster distance. Assume between Cluster i and Cluster j.

Step 2: Merge Cluster j into Cluster i.

Step 3: Delete the distances between Cluster j and other clusters.

Step 4: Update the distances between Cluster i and other clusters using Eq. 3.15.

Step 5: Repeat Steps 2 – 4 until all objects in the same cluster.

The computational complexity for the agglomerative clustering algorithm, $O(n^2)$, is well-established in the literature (Jain and Dubes, 1988).

3.3.2 Evaluation Methodology

In the chapter on literature review, we discussed that internal criteria such as validity indices could be used to discover inherent data structures. However, in this chapter, assuming pre-classified data is provided, we can adopt an external criterion which measures the performance of hierarchical clustering algorithms with various distances against the classes assigned a priori. Clustering accuracy is used as a main external measure of clustering results. Let a_i denote the number of objects correctly assigned to the true class C_i, while b_i denotes the number of objects incorrectly assigned to the true class C_i and c_i the number of objects incorrectly rejected from the true class C_i. Clustering accuracy is defined as

$$d = \frac{1}{n} \sum_{i=1}^{k} a_i. \quad (3.16)$$

Clustering error is defined as $e = 1 - d$. The error e_i for class C_i is $1 - a_i/n$. The precision p_i and recall r_i of class C_i is given by

$$p_i = \frac{a_i}{a_i + b_i} \qquad \text{and} \qquad r_i = \frac{a_i}{a_i + c_i}.$$

The average precision and average recall are $\frac{1}{k}\sum_{i=1}^{k} p_i$ and $\frac{1}{k}\sum_{i=1}^{k} r_i$, respectively. Since $\sum_{i=1}^{k}(a_i + b_i) = \sum_{i=1}^{k}(a_i + c_i) = n$, the average precision, the average recall, and clustering accuracy are the same.

3.4 Experiment

The algorithm will be tested on two kinds of datasets, synthetic and real. The use of constructed artificial datasets allows us to control their structures and facilitates investigation of which distance brings better results in which scenarios.

In order to investigate the performance of the agglomerative hierarchical algorithm integrated with four different distances, namely, the co-occurrence distance, the Goodall distance, the *k*-prototype distance, and the optimal weight distance, we cut the generated trees according to the priori information, the true number of clusters, and then compare the set of generated clusters with the true classes.

The weights in both the *k*-prototype and optimal weighted distances need to be decided. The *k*-prototype distance needs a user-specified weight γ to balance the nominal and numeric distances. As Huang suggested, the weight in *k*-prototype distance, γ, should be in the range of [0.5, 1.4]; therefore, in the experiment, γ will be set as {0.5, 0.7, 0.9, 1.1, 1.3}. We pick the solution with the highest accurate rate. What's more, the weight w in the optimal weight distance needs intensive search in the range of [0, 1]. We therefore set w from 0 to 1 by increasing 0.05 and calculate the ratio of the within-cluster and between-cluster weighted distances given the number of clusters. The best solution is the one with the minimal ratio.

3.4.1 Synthetic Datasets

3.4.1.1 Datasets Description

We create 29 synthetic datasets, labeled *ds1* through *ds29*, to explore various dataset structures. Dataset *ds1* contains two categorical attributes (Cat.1 and Cat.2) and two real attributes (Real.1 and Real.2). There are total 600 instances, equally distributed into three classes, CLS1, CLS2, and CLS3. The categorical attribute values are predefined and assigned to each class in equal proportion. Cat.1 has a unique symbolic value for each class, while Cat.2 has two distinct symbolic values for each class. The real attribute values are generated by sampling normal distributions with different means and standard deviations for each class. The three clusters are well-separated from each other, and thus relatively easy to identify.

Cat.1	Cat.2	Real.1	Real.2	Class	# obs
M1	A2	N(4, 0.3)	N(24, 3)	CLS1	100
	B2				100
F1	C2	N(5, 0.3)	N(26, 3)	CLS2	100
	D2				100
G1	E2	N(6, 0.3)	N(28, 3)	CLS3	100
	F2				100

Table 1: The base dataset *ds1* — three well-separated clusters.

The dataset *ds1* may be considered as the base dataset. To analyze how degrees of co-occurrence relations affect clustering methods, a dataset, *ds2*, is constructed by introducing an additional categorical attribute, Cat.3, into the base dataset. Real.2 is normally distributed with mean of 26 and deviation of 3, but Real.1 is corrupted by introducing 20% noise from CLS3. The value of Cat.3 is assigned according to its Real.1 value. Thus, there is a strong connection between Cat.3 and Real.1, but a weak association between Real.1 and the target class. The datasets *ds3* and *ds4* increase the noise from CLS3 by 40% and 60%, respectively.

Cat.1	Cat.2	Cat.3	Real.1	Real.2	Class	# obs
M1	A2, B2	A3	N(4, 0.3)		CLS1	160
		C3	N(6, 0.3)			40
F1	C2	B3	N(5, 0.3)	N(26, 3)	CLS2	100
	D2					100
G1	E2	C3	N(6, 0.3)		CLS3	100
	F2					100

Table 2: *ds2* — co-occurrence with 20% noise.

A stronger co-occurrence relation is established in the datasets *ds5 – ds7* by associating Real.2 with Cat.3. The means of normal distributions in Real.2 are assigned according to the values of Cat.3. Therefore, Real.1, Real.2, and Cat.3 have a co-occurrence relation. However, the contributions of Cat.3, Real.1, and Real.2 to the target class are weakened. Class 1 is corrupted by introducing noise from Class 3 by 20%, 40%, and 60%, respectively.

Cat.1	Cat.2	Cat.3	Real.1	Real.2	Class	# obs
M1	A2, B2	A3	N(4, 0.3)	N(24, 3)	CLS1	160
		C3	N(6, 0.3)	N(28, 3)		40
F1	C2	B3	N(5, 0.3)	N(26, 3)	CLS2	100
	D2					100
G1	E2	C3	N(6, 0.3)	N(28, 3)	CLS3	100
	F2					100

Table 3: *ds5* — stronger co-occurrence with 20% noise.

The impact of non-Gaussian noise in categorical or real attributes can be observed by randomly picking some categorical or real attribute values in each class and switching them with other classes. First, we randomly choose 20 percent of the instances from each class and change their categorical or real values. Then, we increase the exchange rate to 40% and 60%, respectively. The datasets *ds8 – ds10* are generated by changing their categorical values while *ds11 – ds13* by changing real values.

Based on *ds2* and *ds5*, we apply the same process by exchanging the categorical values and real values of the instances in each class by 20%, 40%, and 60%, respectively. Thus, the datasets *ds14 – ds25* are created.

In order to investigate the effect of real attributes on the clustering results, we relax categorical attribute by assigning them equally (*ds26*). The target class thus, is dependent only on the real attributes and unrelated to the categorical attributes. In *ds27*, on the contrary, the real attribute values are sampled from an identical uniform distribution for all classes. As a result, the target class is dependent only on the categorical attributes and unrelated to the real attributes.

Cat.1	Cat.2	Real.1	Real.2	Class	# obs
M1(⅓),	A2(1/6),	N(4, 0.3)	N(24, 3)	CLS1	200
F1(⅓),	…,	N(5, 0.3)	N(26, 3)	CLS2	200
and G1(⅓)	F2(1/6)	N(6, 0.3)	N(28, 3)	CLS3	200

Table 4: *ds26* — clusters only dependent on real attributes.

Cat.1	Cat.2	Real.1	Real.2	Class	# obs
M1	A2			CLS1	100
	B2				100
F1	C2	U(7, 13)	U(26, 38)	CLS2	100
	D2				100
G1	E2			CLS3	100
	F2				100

Table 5: *ds27* — clusters only dependent on categorical attributes.

The last two datasets (*ds28* and *ds29*) are created by relaxing Cat.1 and Cat.2, respectively. Then the class is related to the two numeric variables and one categorical variable.

Cat.1	Cat.2	Real.1	Real.2	Class	# obs
M1	A2(1/6),	N(4, 0.3)	N(24, 3)	CLS1	200
F1	…,	N(5, 0.3)	N(26, 3)	CLS2	200
G1	F2(1/6)	N(6, 0.3)	N(28, 3)	CLS3	200

Table 6: *ds29* — clusters only dependent on real attributes and Cat.1.

3.4.1.2 Results for Four Distances

The overall accuracy rates of the hierarchical algorithm integrated with four different distances are provided in the following tables (Table 7 – Table 11). For the well-separated base dataset (*ds1*), except the optimal weight distance, all other three have high accuracy. When introducing a new categorical variable (Cat.3) associated with Real.1 and adding some noise, the co-occurrence distance and the *k*-prototype distance still have good performance (see Table 7).

	Occurr.	Goodall	*k*-prot.	Opt. Weight	Description
ds1	100.00%	94.83%	100.00%	79.00%	three well-separated classes
ds2	100.00%	92.83%	100.00%	62.67%	occurrence + 20% noise
ds3	100.00%	86.33%	100.00%	86.67%	occurrence + 40% noise
ds4	100.00%	79.17%	100.00%	66.33%	occurrence + 60% noise
ds5	100.00%	92.67%	100.00%	83.83%	Stronger occurrence + 20% noise
ds6	100.00%	86.33%	100.00%	86.67%	Stronger occurrence + 40% noise
ds7	100.00%	79.67%	100.00%	80.00%	Stronger occurrence + 60% noise

Table 7: Accuracy of synthetic datasets with co-occurrence.

The Goodall distance is stable when introducing categorical non-Gaussian noise. Although the co-occurrence distance and the *k*-prototype distance are insensitive to numeric non-Gaussian noise, their performance deteriorates quickly when a large amount of categorical non-Gaussian noise is introduced, as can be seen in the result of *ds10* in Table 8.

	Occurr.	Goodall	*k*-prot.	Opt. Weight	Description
ds8	80.00%	88.50%	80.00%	64.33%	20% cat. non-Gaussian noise
ds9	60.00%	78.33%	60.00%	79.83%	40% cat. non-Gaussian noise
ds10	40.00%	72.83%	40.00%	63.33%	60% cat. non-Gaussian noise
ds11	100.00%	73.33%	100.00%	61.83%	20% real non-Gaussian noise
ds12	100.00%	57.83%	100.00%	86.67%	40% real non-Gaussian noise
ds13	100.00%	40.67%	100.00%	35.33%	60% real non-Gaussian noise

Table 8: Accuracy of synthetic datasets when adding non-Gaussian noise.

As can been seen in Table 9 and Table 10, when the categorical non-Gaussian noise is applied to the dataset having a co-occurrence relation between some attributes (e.g., Cat.3, Real.1, and Cat2.), the results become worse as the noise increases. On the other hand, the four distances are all good at handling real non-Gaussian noise, especially the co-occurrence distances and the *k*-prototype distances.

	Occurr.	Goodall	*k*-prot.	Opt. Weight	Description
ds14	80.00%	76.83%	80.00%	75.33%	20% noise + Occur. +20% cat. non-Gau
ds15	60.00%	61.33%	60.00%	62.00%	20% noise + Occur. +40% cat. non-Gau
ds16	40.00%	41.67%	40.00%	54.50%	20% noise + Occur. +60% cat. non-Gau
ds17	80.00%	76.00%	80.00%	78.33%	20% noise+S. Occur.+20% cat. non-Gau
ds18	60.00%	58.00%	60.00%	70.17%	20% noise+S. Occur.+40% cat. non-Gau
ds19	40.00%	31.33%	40.00%	60.50%	20% noise+S. Occur.+60% cat. non-Gau

Table 9: Accuracy of datasets with co-occurrence & nominal non-Gaussian noise.

	Occurr.	Goodall	*k*-prot.	Opt. Weight	Description
ds20	100.00%	87.83%	100.00%	87.83%	20% noise+Occur. + 20% real non-Gau
ds21	100.00%	85.00%	100.00%	100.00%	20% noise+Occur. + 40% real non-Gau
ds22	100.00%	88.17%	100.00%	99.50%	20% noise+Occur. + 60% real non-Gau
ds23	100.00%	91.17%	100.00%	57.13%	20% noise+S. Occur.+ 20% real non-Gau
ds24	100.00%	87.33%	100.00%	87.17%	20% noise+S. Occur.+ 40% real non-Gau
ds25	100.00%	85.50%	100.00%	42.17%	20% noise+ S. Occur.+60% real non-Gau

Table 10: Accuracy of datasets with co-occurrence & real non-Gaussian noise.

When relaxing some attributes, leading to unrelated or redundant information, the four distances show different behaviors. As can be seen in the results of *ds26*, the co-occurrence distance is not good at discriminating real attributes, but excellent at categorical attributes (as shown in the result of *ds27*). Even one categorical attribute clearly identifying the class would result in a high accuracy. On the contrary, the *k*-prototype distance has a very low accuracy when relaxing the categorical attribute(s). The Goodall and optimal weight distances seem to be confused by the redundant information and lead to poor performance.

	Occurr.	Goodall	*k*-prot.	Opt. Weight	Description
ds26	33.83%	76.33%	31.67%	71.33%	Relax categorical variables.
ds27	100.00%	35.17%	100.00%	62.00%	Relax numeric variables.
ds28	100.00%	87.83%	36.67%	70.17%	Relax Cat. 1
ds29	100.00%	83.00%	36.50%	60.50%	Relax Cat. 2

Table 11: Accuracy of synthetic datasets when relaxing some attributes.

From the synthetic study, the optimal weight distance is relatively unstable in comparison with other three. Categorical non-Gaussian noise has great impact on the four distances. Their performance deteriorates quickly as the noise increases. On the other hand, the four distances are all good at handling real non-Gaussian noise, especially the co-occurrence distances and the *k*-prototype distances. The co-occurrence distance is not good at discriminating real attributes, but excellent at categorical attributes. Even one categorical attribute clearly identifying the class would result in a high accuracy. The *k*-prototype distance has a very low accuracy when relaxing the categorical attribute(s).

3.4.2 Real-world Datasets

3.4.2.1 Datasets Description

While informative, the constructed datasets may not well represent real-world data. Therefore, we chose six datasets from UCI ML repository (http://www.sgi.com/tech/mlc/db), as shown in Table 12, two datasets with mixed types (Heart Disease, Australian Credit), one with pure numerical attributes (Iris) and three with pure categorical attributes (Iris-Disc, Vote, and DNA-nominal). These datasets are either benchmark datasets widely used in machine learning research community or standard test beds for clustering mixed datasets.

Dataset	# categorical var.	# numeric var.	# obs.	# Class
Heart Disease	8	5	303	2
Iris	0	4	150	3
Iris-Disc	4	0	150	3
Vote	16	0	435	2
Australian Credit	8	6	690	2
DNA-nominal	60	0	3,186	3

Table 12: Six real datasets from UCI.

The Heart Disease data is generated at the Cleveland Clinic. It records 164 normal people and 139 heart patients. These 303 instances contain eight nominal and five continuous attributes, along with the class label: no heart disease or with different degrees of heart disease. In the preprocess procedure, six missing values are replaced with the modes of corresponding classes. Five continuous features are discretized into five equal-width intervals.

The Iris dataset contains three classes of 50 instances each. The four numeric-valued attributes describe the sepal length, sepal width, petal length, and petal width of each plant. One type of Iris is linearly separable from the other two, but the latter are mixed. All the attributes are discretized into five equal-width intervals.

The Iris-Disc dataset is the Iris dataset discretized using some functions in MLC++ discretize utility which exploit some optimal techniques for attribute discretization. Sepal width is converted to two levels and all other attributes to three levels.

The Congressional Votes dataset is the United States Congressional Voting Records in 1984. Each record corresponds to one congressman's votes on 16 issues (e.g., education spending, crime). Class label (Republican/Democrat) is provided for each instance. There are 168 Republicans and 267 Democrats, for a total of 435.

The Australian Credit dataset has six numeric and eight categorical attributes. Among 690 instances, 307 instances came from approved applications and 383 from rejected applications. This dataset includes the nominal features with not only small numbers of values, but also large numbers of values, e.g., 14-value and 8-value attributes. In data

preprocess phase, we combine some levels and get four values for each attribute. The numeric features are discretized into intervals with equal width.

The DNA-nominal dataset has 3,186 primate splice-junction gene sequences described by 60 nucleotides. These 60 attributes are represented by A, C, G, or T. A class label – acceptor, donor, or non-splice – is attached with each instance. The dataset contains records for 767 acceptors, 765 donors, and 1654 that are neither of them.

3.4.2.2. Results for Four Distances

The results of the hierarchical algorithm with four distances are displayed in Table 13, and the corresponding weights of the *k*-prototype and optimal weight distances are provided. The bold font highlights the best solution for each dataset.

	Co-occurrence	Goodall	*k*-prototype	Optimal Weight
Heart	76.24%	77.23%	**81.52%** $\gamma = 0.7$	78.55% $w = 0.80$
Iris	88.67%	**90.00%**	88.67%	82.67%
Iris-Disc	94.67%	**96.00%**	**96.00%**	**96.00%**
Vote	84.60%	**91.72%**	86.21% $\gamma = 0.9$	84.37%
Aus-Credit	**84.78%**	71.74%	83.48% $\gamma = 0.9$	60.15% $w = 0.65$
DNA	**81.33%**	73.98%	78.15% $\gamma = 0.9$	76.15%

Table 13: Accuracy of real datasets with four distances.

Although the co-occurrence distance is not always better than other distances when used with hierarchical clustering, the proposed algorithm has some advantages over the other distance representations. First, the running time to calculate the co-occurrence distance is not as prohibitive as to obtain the Goodall distance since the latter costs a polynomial in the unique values of the numeric attributes. This number is much larger than the level of categorical attributes used in the time complexity of the co-occurrence distance calculation. Second, it does not require the number of clusters to decide the weight that balances the nominal and numeric distances. As can be seen, the weights of the *k*-prototype and the optimal weighted distances have a great impact on the performance of the hierarchical algorithms. In order to achieve the best solution, the hierarchical algorithms need different weights for the partitions with different number of clusters.

Under the assumption that no priori information such as the number of clusters is available, there are only two distances are applicable, namely, the co-occurrence and the

Goodall distances. Continuing on the second part of the proposed algorithm for searching the proper number of clusters, we only apply the co-occurrence distance to the agglomerative hierarchical clustering because of the time complexity.

3.4.2.3 Comparative Study for Heart Dataset

Since most mixed clustering algorithms use the Heart dataset to test their performance, this proposed clustering algorithm – hierarchical clustering with the co-occurrence distance – is also applied on this dataset and compared to other five mixed clustering algorithms, which come from two categories, namely, the hierarchical and the *k*-means families. The SBAC and ECOWEB are from the first category, while the *k*-prototype, the optimal weight *k*-means, and the fuzzy prototype *k*-means are from the second.

Method	Recovery Matrix		Class Error	Accuracy
Hierarchical on	101	34	25.19%	76.24%
co-occurrence distance	38	130	22.62%	
SBAC	126	37	22.70%	75.25%
	38	102	27.14%	
ECOWEB	105	20	16.00%	73.93%
	59	119	33.15%	
k-prototype	116	55	32.16%	66.01%
	48	84	36.36%	
Optimal Weight	136	32	19.05%	80.20%
k-means(w=0.91)	28	107	20.74%	
Fuzzy prototype	139	21	13.13%	84.82%
k-means	25	118	17.48%	

Table 14: Comparative study on Heart Disease dataset.

As can be seen in Table 14, except the optimal weight *k*-means and the fuzzy prototype *k*-means, the results are better than the other algorithms. However, if there are no priori information on the number of clusters, *k*-means algorithms may not be applicable since they require the number of cluster as an input parameter.

3.4.2.4. The Results of Co-occurrence Distance

The details of results for the hierarchical clustering with Ward's linkage based on the co-occurrence distance for real-world datasets are presented in Table 15. It includes

clustering recovery matrix, clustering accuracy and the error for each class. These results demonstrate this proposed clustering is capable of partitioning mixed datasets with a high rate of accuracy. The accuracy for Iris-Disc is higher than for Iris because the conversion from real attributes to categorical attributes exploits some optimal techniques to search the best splitting strategy and results in homogeneous intervals. Accordingly, the Iris-Disc values of the categorical attributes for the same type of the plants would share the same new assigned label with a large probability. Therefore, the procedure of optimal discretiztion enhances the co-occurrence between attributes, which improves the performance and gets better results than the general discretization using equal-width intervals as in Iris.

Dataset	Recovery Matrix			Class Error	Accuracy
Heart Disease	101	34		25.19%	76.24%
	38	130		22.62%	
Iris	50	0	0	0.00%	88.67%
	0	50	17	25.37%	
	0	0	33	0.00%	
Iris-Disc	50	0	0	0.00%	94.67%
	0	49	7	12.50%	
	0	1	43	2.27%	
Vote	161	60		27.15%	84.60%
	7	207		3.27%	
Aus-Credit	321	43		11.81%	84.78%
	62	264		19.02%	
DNA	502	30	110	21.81%	81.33%
	133	691	146	28.76%	
	132	44	1398	11.18%	

Table 15: Results of real datasets with the co-occurrence distance.

3.5 Properties of Proposed Clustering Method

From the study on synthetic datasets, we acquired some insights on the characteristics of datasets, for which the co-occurrence distance would outperform the three other distances when used with the agglomerative hierarchical clustering method. In this section, we further apply these properties and exploit these advantages to preprocess the six real datasets. In doing so, the result dataset would be more applicable to the co-occurrence distance when used with hierarchical clustering.

3.5.1 Iris Dataset

The attributes sepal length (SL) and sepal width (SW) of an Iris plant are very difficult to use to separate Class 2 and Class 3, which are represented by "+" and "Δ" in Figure 2, respectively. However, from the scatter plot of petal length (PL) versus petal width (PW), though Class 2 and Class 3 have a little overlap, they are much easier to identify.

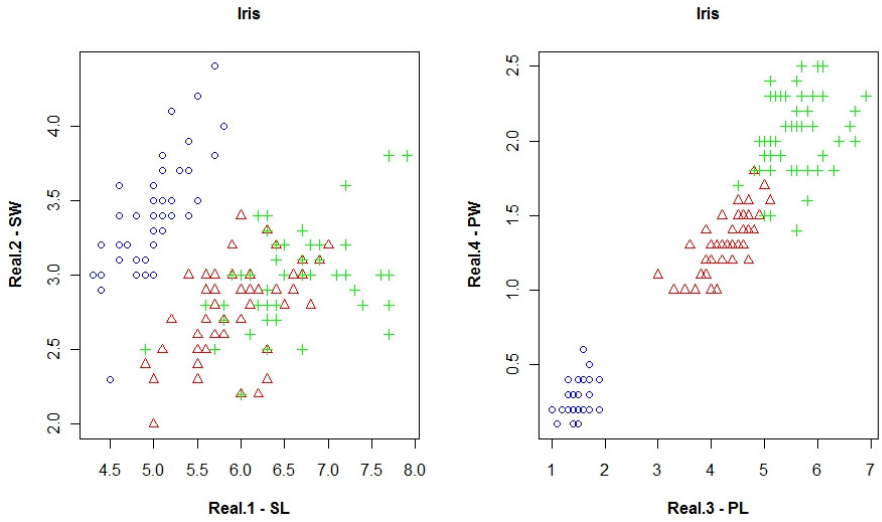

Figure 2: The scatter plots of Iris. (Left) SL vs. SW. (Right) PL vs. PW.

Choosing the discretized PL and PW, the co-occurrence distance will generate best results among the four distances, as seen the result of datasets Iris 2 and Iris 5 in Table 16. From the synthetic data study, the co-occurrence distance gives more weights on categorical attributes, and is not able to handle the categorical non-Gaussian noise well, but performs well with the redundant and noisy real attributes.

	Occurr.	Goodall	*k*-prototype	Opt. Weight	Description
Iris	88.67%	**90.00%**	88.67%	82.67%	
Iris 1	71.33%	**97.33%**	73.33%	96.67%	Discretize SW and SL; PW, PL
Iris 2	**95.33%**	92.67%	**95.33%**	**95.33%**	Discretize PW and PL; SW and SL
Iris 3	94.67%	**95.33%**	94.67%	94.67%	Four real and four categorical Attr.
Iris 4	65.33%	**71.33%**	58.67%	58.67%	Discretize SW and SL
Iris 5	**96.00%**	92.67%	95.33%	95.33%	Discretize PW and PL

Table 16: Accuracy of preprocessed Iris dataset compared to original.

3.5.2 Vote Dataset

The Vote dataset includes the attributes that clearly identify the class, e.g. Cat. 4, as well as very noisy attributes, e.g. Cat.2. We pick up the attributes according to their significance calculated by the co-occurrence and random forest methods.

First choose most significant attributes: C3 and C4, and then add the less significant one. In order to test the effect the least significant attributes, the nine least significant attributes (C1, C2, C9, C10, C11, C12, C13, C15, and C16) form dataset Vote 5. In addition, we select the four most significant attributes among the weakest (C9, C12, C13, and C16) in comparison.

	Occurr.	Goodall	*k*-prototype	Opt. Weight	Description
Vote	84.60%	**91.72%**	86.21%	84.37%	
Vote 1	**95.63%**	90.81%	87.36%	87.36%	Significant Attr. (C3, C4)
Vote 2	**87.59%**	86.67%	86.67%	86.67%	Significant Attr. (C4, C8)
Vote 3	**91.95%**	90.81%	**91.95%**	**91.95%**	Significant Attr. (C3, C4, C5)
Vote 4	85.52%	85.75%	**87.36%**	85.75%	Significant Attr. (C3, C4, C5, C8)
Vote 5	83.68%	82.99%	**87.59%**	**87.59%**	The nine weakest Attr.
Vote 6	80.23%	82.76%	**84.14%**	**84.14%**	The four Attr. among the weakest

Table 17: Accuracy of preprocessed Vote dataset compared to original.

The co-occurrence distance can improve performance when choosing the most significant categorical attributes, but does not handle very well with noisy categorical data as shown by Vote 5 and Vote 6 in Table 17.

3.5.3 Heart Disease Dataset

There are no significant individual categorical attributes. Both categorical and real attributes are noisy. To begin with, rank the attributes by the significance calculated with the co-occurrence method. The two most important categorical and real attributes form the dataset Heart 1, and then the first five most important attributes are chosen as dataset Heart 2. Next scenario includes the nine most important attributes. The latter two are made of only one data type. The result shows only considering a few important variables the performance of the co-occurrence method is no less than other measures.

	Occurr.	Goodall	*k*-prototype	Opt. Weight	Description
Heart	77.23%	77.23%	**81.52%**	78.55%	
Heart 1	**76.57%**	67.99%	**76.57%**	75.91%	Significant Attr. (C4, C8, R3, R5)
Heart 2	**76.57%**	76.24%	**76.57%**	**76.57%**	Significant Attr. (C4, C8, R3, R4, R5)
Heart 3	71.29%	71.62%	**76.24%**	70.96%	C2, C4, C5, C8, R1, R2, R3, R4, R5
Heart 4	76.24%	75.58%	**78.88%**	**78.88%**	All categorical attributes
Heart 5	**72.61%**	66.34%	61.06%	65.68%	All numeric attributes

Table 18: Accuracy of preprocessed Heart Disease dataset compared to original.

3.5.4 Australian Credit Dataset

One of categorical attributes, C5, is able to clearly identify the class in Australian Credit dataset, but many categorical attributes are noisy. The first two are real attributes, R3 and R4, in the order of the significance of attributes by the co-occurrence relation. We add the significant categorical attributes one by one, until get a reasonable accuracy, seen the datasets Aus 1 and Aus 2. Remove the two real attributes to generate the dataset Aus 3, the result are what we expected.

	Occurr.	Goodall	*k*-prototype	Opt. Weight	Description
Aus	**84.78%**	71.74%	83.48%	60.15%	
Aus 1	73.19%	73.62%	**83.19%**	73.48%	C2, C4, C5, C6, C8, R3, R4
Aus 2	**85.51%**	73.23%	77.97%	73.48%	C2, C3, C4, C5, C6, C8, R3, R4,
Aus 3	**85.51%**	53.19%	84.20%	84.20%	C2, C3, C4, C5, C6, C8
Aus 4	**85.22%**	82.99%	84.78%	83.19%	All categorical attributes
Aus 5	62.75%	**68.12%**	63.62%	66.67%	All real attributes

Table 19: Accuracy of preprocessed Australian Credit dataset.

The co-occurrence distance is robust with the redundant and noisy real attributes. The results of datasets Aus 4 and Aus 5 show us that the total categorical attributes make greater contribution to discriminate the class than the total real attributes.

3.5.5 DNA-nominal Dataset

The DNA dataset has sixty four-level categorical attributes to determine the interface of a gene sequence. No categorical attribute could clearly identify the class. Some class has only one level on a particular attribute. For instance, though on attribute 29 all classes have level A, only donors have this one level. Acceptors only have one level G on attribute 31. In

a similar way, these attributes are ranked by their significance calculated with the co-occurrence method. We select three elbow points on the plot of the ordered significance. The first scenario is the twelve most significant attributes. The second is the thirty-eight most important attributes. The last one has all attributes except the four weakest, C1, C53, C55, and C57. The results show after some data preprocessing procedures by choosing the most important attributes, the co-occurrence distance would achieve a higher rate of accuracy.

	Occurr.	Goodall	*k*-prototype	Opt. Weight	Description
DNA	**81.33%**	73.98%	78.15%	76.15%	
DNA 1	77.09%	77.62%	**83.46%**	81.14%	First 12 most significant Attr.
DNA 2	**87.70%**	78.59%	84.02%	84.02%	First 38 most significant Attr.
DNA 3	**82.58%**	78.25%	78.31%	74.33%	All Attr. except C1,C53,C55,C57

Table 20: Accuracy of preprocessed DNA-nominal dataset compared to original.

The properties of the proposed clustering are as follows.

1. It gives categorical attributes more weights than the real ones.
2. It is confused by the noise of other categorical attributes.
3. It would have better performance if one categorical attribute would clearly identify the class.
4. It is robust with redundant or noisy real attributes.

3.6 Summary

In this chapter, we propose an agglomerative hierarchical clustering using the co-occurrence distance to partition the datasets with mixed data types. The performance is tested on a series of synthetic datasets and six standard real-world datasets, and furthermore, compared with other extant distances that would represent both real and categorical
The advantage and disadvantage of the proposed method are presented. Another important advantage of the proposed algorithm is, if the number of clusters is absent, by using the derived tree structure, we can search for the optimal number of clusters. Another reason for using the co-occurrence distance rather than others is that the computational complexity of the Goodall distance calculation is prohibitive, and the weights balancing the nominal and numeric distances in both the *k*-prototype distance and the optimal weight distance change as the number of clusters varies.

CHAPTER 4 *BK* INDEX FOR MIXED DATA

The output of any hierarchical clustering method such as the new variant proposed in Chapter 3, is a dendrogram or a tree structure from which a set of clusters can be obtained. However, this set of clusters is not unique, another important procedure is to determine the optimal points where cut the dendrogram to form a set of clusters. While this problem, which is usually referred to as cluster validation is well-known, there are few indices that can handle a dataset with both categorical and numeric attributes. The Calinski-Harabasz (*CH*) index defined in Chapter 2 has shown outstanding performance in the numeric domain. It was the best among the top 30 indices ranked by Milligan and Cooper (1985). The *CH* index searches the proper number of clusters by maximizing the ratio of the between-cluster and within-cluster scatter matrices. For categorical datasets, on the other hand, cluster entropy and categorical utility are frequently used.

In this chapter we extend a validity index to find an optimal number of clusters in mixed datasets and integrate it with the hierarchical algorithm proposed in Chapter 3. The proposed validity index will be tested on several datasets, both synthetic and real; and then compared to three other indices that could be extended to handle mixed datasets.

4.1 Motivation

Cluster validation methods are able to evaluate the result clusters quantitatively and objectively, e.g., whether the cluster structure is meaningful or just an artifact of the clustering algorithm. There are two main categories of testing criteria, known as external indices and internal indices dependent on the present of priori information of known categories.

Internal indices are validation measures which evaluate clustering results using only information intrinsic to the underlying data. Without true cluster labels, estimating the number of clusters, k, in a given dataset is a central task in cluster validation.

Although a large number of validity criteria could be used to estimate the number of clusters in pure numeric data or pure categorical data, no index exists to deal directly with the cluster validation problems related to data containing both categorical and numeric attributes.

A family of cluster validation indices exploits inherent geometry or density to discover the underlying structures of numeric datasets. Recall that we generated a geometric-like distance combining both numeric and categorical distances in the preceding chapter. Thus, we could explore extant numeric indices with this geometric-like distance. Three indices are chosen based on their performance and usage reported in the literature review, namely, the Calinski-Harabasz index (*CH*), the Dunn index (*DU*), and the Silhouette index (*SI*). The expected entropy of the partition structure is another way to evaluate the quality of a clustering result, since a low entropy indicates a high ordered structure. Chen and Liu (2009) exploited this property and designed an index called the *BK* index for categorical datasets. In this chapter, we will extend the *BK* index for mixed datasets. Unlike Chen and Liu's algorithm, our approach reduces a computational burden without repeated calculations of cluster entropies.

4.2 Background

4.2.1 Calinski-Harabasz Index

The Calinski-Harabasz index (Calinski and Harabasz, 1974) calculates the ratio of the between-cluster scatter matrix ($\mathbf{S_B}$) and the within-cluster scatter matrix ($\mathbf{S_W}$). It is formulated as,

$$CH(k) = \frac{Tr(\mathbf{S_B})/(k-1)}{Tr(\mathbf{S_W})/(n-k)}, \tag{4.1}$$

where n is the number of objects and k the number of clusters. $Tr(\mathbf{S_B})$ and $Tr(\mathbf{S_W})$ are the traces of the between-class and the within-class scatter matrices, respectively. The formulations are given as follows.

$$Tr(\mathbf{S_B}) = \sum_k n_i \left\| z_i - z \right\|^2 \qquad \text{and} \qquad Tr(\mathbf{S_W}) = \sum_k \sum_i \left\| o_i - z_k \right\|^2,$$

where o_i is an object in the class C_k, z_k the centroid of cluster C_k, and z the centroid of all objects. Since well-separated and compact clusters are desirable, $Tr(\mathbf{S_B})$ is maximized and $Tr(\mathbf{S_W})$ minimized. The value of k that maximizes the *CH* index suggests an estimation of the optimal number of clusters.

4.2.2 Dunn Index

The Dunn index (Dunn, 1974) attempts to identify the clusters that are compact and well-separated by maximizing the inter-cluster distance while minimizing the intra-cluster distance. The Dunn index for k clusters is given as

$$DU(k) = \min_{i=1,\cdots,k}\left(\min_{j=i+1,\cdots,k}\left(\frac{D(C_i, C_j)}{\max_{m=1,\cdots,k} diam(C_m)}\right)\right). \tag{4.2}$$

$D(C_i, C_j)$ is the cluster distance between C_i and C_j and is found by taking the minimum distance between a pair of objects, one object in each cluster.

$$D(C_i, C_j) = \min_{o_i \in C_i, o_j \in C_j} d(o_i, o_j)$$

The diameter of cluster C_m, $diam(C_m)$, is the maximum distance between two objects in a cluster.

$$diam(C_m) = \max_{o_i, o_j \in C_m} d(o_i, o_j)$$

The most probable number of clusters is obtained at the largest value of the Dunn index. One of the disadvantages of the Dunn index is its sensitivity to noise.

4.2.3 Silhouette Index

The Silhouette index (Kaufman and Rousseeuw, 1990) is an average of the silhouette width over all objects. The silhouette width of the i_{th} object is defined as,

$$SI_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{4.3}$$

For the i_{th} object, let a_i be the average distance between object i and the other objects in its own cluster and b_i the minimum of the average distances between the i_{th} object and the other objects in other clusters. Therefore, the silhouette index is determined by

$$SI(k) = \frac{1}{n}\sum_{i=1}^{n}\frac{(b_i - a_i)}{\max(b_i, a_i)}. \tag{4.4}$$

Since the objective is to obtain the clusters with minimum intra-cluster distance and maximum inter-cluster distance, high values for SI are desirable. Thus, the partition with the highest $SI(k)$ is taken to be optimal.

4.3 Proposed Entropy-based Validity

Entropy-based method computes the expected entropy of a partition. The smaller the expected entropy, the better quality of the partition is. The expected entropy decreases monotonically as the number of clusters increases, but from some point onwards the decrease flattens remarkably. Rather than searching for the location of an "elbow" on the plot, Chen and Liu (2009) calculated the second order difference of information gain, which is called the *BK* index.

This index is applied on categorical attributes. In their study, they employed a hierarchical algorithm where cluster distance was the entropy difference between two clusters. This algorithm is computationally expensive since the entropies need to be iteratively calculated for categorical data. Calculating the entropies for numeric attributes is even more problematic because it is difficult to compute and bound density estimates of numeric attributes. In this chapter, therefore, we will employ the proposed hierarchical algorithm in the preceding chapter to identify the optimal number of clusters determined by the internal validation criterion.

4.3.1 Notation

The entropy for a dataset $DS(U, A)$ is defined as

$$\mathrm{H}(DS) = -\sum_{a_i \in A} \sum_{s \in V_{a_i}} \Pr(a_i(x) = s) \log \Pr(a_i(x) = s). \tag{4.5}$$

If a dataset is partitioned into a set of groups P_k, $(C_1, \cdots, C_k)$, then the entropy for an individual C_j is

$$\mathrm{H}(C_j) = -\sum_{a_i \in A} \sum_{s \in V_{a_i}} \Pr(a_i(x) = s \mid x \in C_j) \log \Pr(a_i(x) = s \mid x \in C_j). \tag{4.6}$$

The expected entropy of a partition P of DS with k clusters is

$$\mathrm{EE}(P_k) = \frac{1}{n} \sum_{C_j \in P} n_j \mathrm{H}(C_j), \tag{4.7}$$

where n_j is the number of objects in C_j.

The entropy measures the prior uncertainty or impurity in a dataset. A small value indicates an ordered cluster structure. In general, it is expected each cluster to be pure, which means that objects in the same cluster come from a single class rather than from different classes. An optimal partition with k clusters that minimizes the expected entropy can be obtained by solving the LP problem as follows:

$$\mathrm{H}_{opt}(k) = \min_{P(k)}\left(\frac{1}{n}\sum_{C_j \in P(k)} n_j \mathrm{H}(C_j)\right). \tag{4.8}$$

Chen and Liu (2009) showed that $\mathrm{H}_{opt}(k)$ satisfies the following two properties:

(1) $\mathrm{H}_{opt}(k) \in [0, \mathrm{H}(DS)]$

(2) $\mathrm{H}_{opt}(k)$ is non-increasing with respect to k.

If each object forms a singleton cluster, then the dataset is divided into n clusters. Thus, $\mathrm{H}_{opt}(n) = 0$. On the other hand, if all objects gather into one cluster, then $\mathrm{H}_{opt}(1) = \mathrm{H}(DS)$. When k increases, the optimal partition $P(k)$, which has the minimal expected entropy, will tend toward a greater ordered configuration, thus, a lower entropy. When k achieves the correct number of clusters, increasing k on a small range will not change the current configuration dramatically, but rather tune the structural order, e.g., splitting a subclass.

4.3.2 *BK* Index

Chen and Liu (2009) define the *BK* index as the second order difference of incremental expected entropy of the partition structure. The highest value of the *BK* index indicates the potential number of clusters.

Let's define the information gain $\mathrm{I}(k)$ as the entropy difference between $\mathrm{H}_{opt}(k)$ and $\mathrm{H}_{opt}(k+1)$. In other words, $\mathrm{I}(k)$ tells us how much would be gained by fitting the data from $k+1$ clusters to k clusters. $\mathrm{B}(k)$ is the second order difference of $\mathrm{I}(k)$.

$$\mathrm{I}(k) = \mathrm{H}_{opt}(k) - \mathrm{H}_{opt}(k+1) \tag{4.9}$$

$$\begin{aligned}\mathrm{B}(k) &= \Delta^2\mathrm{I}(k) = \Delta\mathrm{I}(k-1) - \Delta\mathrm{I}(k) \\ &= (\mathrm{I}(k-1) - \mathrm{I}(k)) - (\mathrm{I}(k) - \mathrm{I}(k+1))\end{aligned} \tag{4.10}$$

Assume that the optimal $P(k)$, $P(k+1)$, and $P(k+2)$ have similar configurations. For instance, $(C_1, \cdots, C_{k-1})$ in $P(k)$ are similar to $(C'_1, \cdots, C'_{k-1})$ in $P(k+1)$, respectively. If the proportions of C'_k and C'_{k+1} are the same, then the combination of C'_k and C'_{k+1} will not increase the expected entropy. Thus, $I(k) = 0$. If the two clusters have similar structures, then the combination will increase the entropy, but by a small amount. The information gain $I(k)$ is small. In a similar way, if $P(k+2)$ has a structure similar to $P(k+1)$, the information gain $I(k+1)$ is also small when converting the optimal partition $P(k+2)$ to $P(k+1)$. On the contrary, if the structure of $P(k-1)$ is different from that of $P(k)$, then fitting the data from k clusters into k-1 clusters would generate a larger information gain $I(k-1)$ since the proportion of each class is seriously disturbed and leads to more impurity in $P(k-1)$. If the configuration of partition $P(k-1)$ changes significantly in comparison with $P(k)$, but keeps stable in successive numbers after k, then k would be among the candidates for the optimal number of clusters. In this case, $B(k)$ is large. To decide the best k, we can calculate the indices from two upwards and pick the k with the largest $B(k)$.

The question of deciding on the best k for mixed data clustering seems to be solved. However, as mentioned in the chapter on literature review, it is NP-hard to attack the problem of finding $H_{opt}(k)$. As a result, some efficient approximate approaches should be adopted to solve this optimization problem. In the preceding chapter, the experiment showed the proposed algorithm, in comparison to the true classes, could recover the cluster structure with high accuracy. We can employ the tree structure constructed by this hierarchical algorithm to tackle the optimization problem of Eq. 4.8.

4.3.3 Proposed Algorithm

We still use the agglomerative hierarchical algorithm incorporating the co-occurrence distance, which is proposed in Chapter 3, to calculate the *BK* index that for both numeric and nominal attributes. However, the *BK* index first needs to be extended to numeric attributes. To avoid prohibitive entropies calculations for numeric attributes, we use the discretized dataset $DS(U', A')$ rather than the original mixed datasets $DS(U, A)$. The proposed algorithm to evaluate the correct number of clusters is described as follows.

INPUT. A dendrogram and the discretized dataset $DS(U', A')$.

OUTPUT. The most probable number of clusters.

Initial: Compute the entropy of the whole dataset and k ← 2.

Step 1: Calculate the expected entropy of a partition EE(P_k), EE(P_{k+1}), and EE(P_{k+2}).

Step 2: Calculate *Information Gain* values I(*k*-1), I(*k*) and I(*k*+1) using Eq. 4.9.

Step 3: Obtain B(*k*) using Eq. 4.10.

Step 4: Increase *k* by one, and repeat Steps 1 – 3 until reaching some stop criterion.

Step 5: Find the maximum B(*k*) and return the corresponding *k*.

4.4 Experiment

In order to test the proposed index, we continue on the synthetic and real-world datasets described in Chapter 3. Using the corresponding trees derived in the preceding chapter, we first calculate the validity indices, B(*k*), with respect to *k* from two up to 18, and then compare our proposed numbers with the correct numbers of true groups, along with other three indices, namely, the Calinski-Harabasz index (*CH*), the Dunn index (*DU*), and the Silhouette index (*SI*).

4.4.1 Synthetic Datasets

There are three classes in all synthetic datasets. Recall the first one is the base dataset (*ds1*) with three well-separated classes. By setting a co-occurrence relation in the attributes and introducing noise from Class 3 to Class 1, the datasets *ds2* – ds*4* are generated. Further a stronger co-occurrence relation appears in the datasets *ds5* – ds*7*. The datasets *ds8 – ds13* are generated by introducing non-Gaussian noise on categorical and real attributes respectively on *ds1*. Based on *ds2* and *ds5*, the datasets *ds14 – ds19* only add categorical non-Gaussian noise while the datasets *ds20 – ds25* add real non-Gaussian noise. The last four relax some particular attribute(s). The detail of each dataset could be seen in Section 3.4.1.1.

The estimated numbers of clusters of the four validity indices are presented in Table 21 and Table 22. The bold font indicates the number equal to the true one. –‘s represent the method is invalid for evaluation. For the base dataset (*ds1*), except the *CH* index, other three indices obtain the correct number of groups, which can be observed at the maximum points in the plots of four indices with respect from 2 to 18 in Figure 3. However, when adding the co-occurrence relation in the attributes of the dataset, only the *BK* index can detect the proper

number when Class 1 includes 20% and 40% noise from Class 3. When the noise increases up to 60%, the *BK* index fails to evaluate the number, as seen in *ds4* and *ds7*. Figure 4 illustrates the *BK* index confuses the number among 2, 3, and 4 for the very noisy dataset *ds4*.

	BK	***CH***	***DU***	***SI***	**Description**
ds1	**3**	**-**	**3**	**3**	three well-separated classes
ds2	**3**	**-**	4	4	occurrence + 20% noise
ds3	**3**	**-**	4	4	occurrence + 40% noise
ds4	2	**-**	4	4	occurrence + 60% noise
ds5	**3**	**-**	4	4	Stronger occurrence + 20% noise
ds6	**3**	**-**	4	4	Stronger occurrence + 40% noise
ds7	2	**-**	4	4	Stronger occurrence + 60% noise

Table 21: Estimated numbers of clusters by four validity indices.

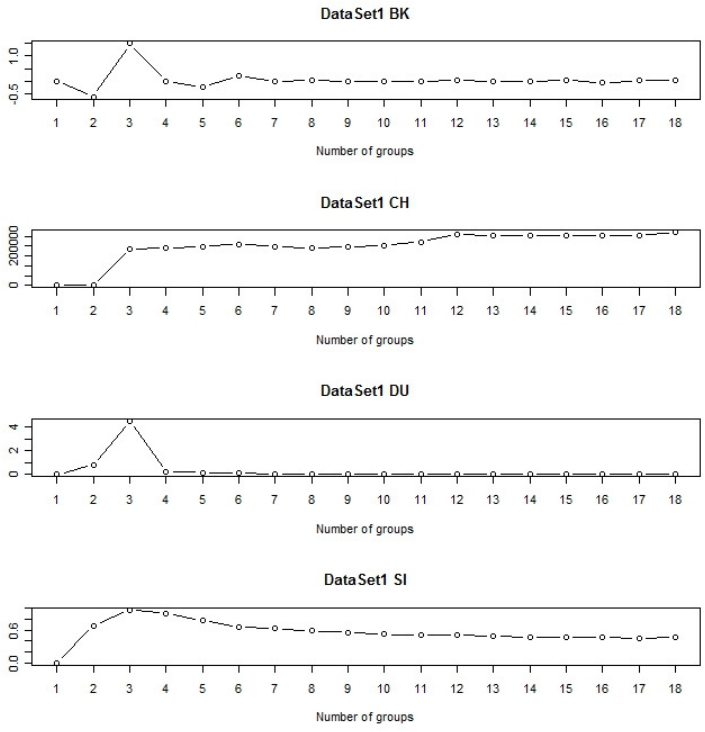

Figure 3: Plots of four indices on base dataset *ds1*.

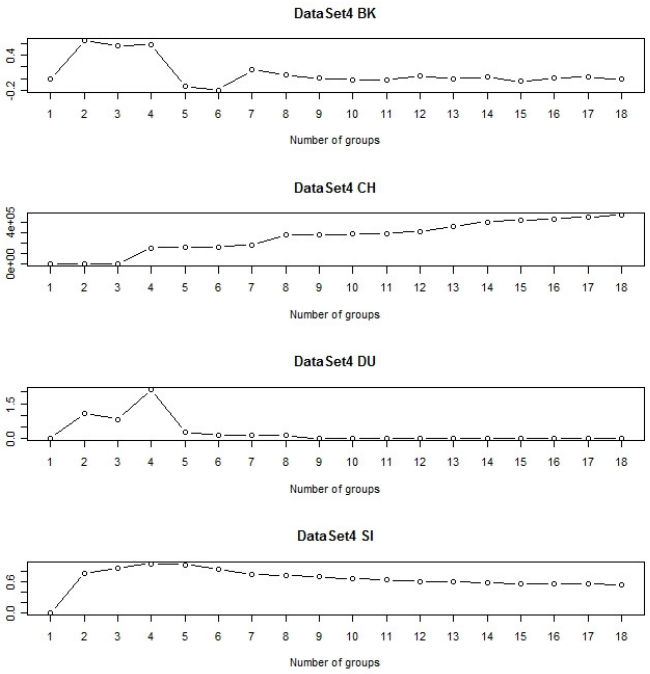

Figure 4: Plots of four indices on very noisy dataset *ds4*.

When adding the non-Gaussian noise by switching the data in different classes, the results keep the same in *ds8 – ds25* in comparison with their base datasets. The *BK*, *DU*, and *SI* indices find the correct number of groups in *ds8 – ds13*. Unfortunately, in *ds14 – ds25*, the *DU* and *SI* indices obtain four instead of three.

	BK	***CH***	***DU***	***SI***	**Description**
ds8 – 13	**3**	**-**	**3**	**3**	Adding non-Gaussian noise
ds14 – 19	**3**	-	4	4	20% noise+Occur.+cat. non-Gaussian
ds20 – 25	**3**	-	4	4	20% noise+Occur.+real non-Gaussian
ds26	4	-	-	-	Relax categorical variables
ds27	**3**	-	**3**	**3**	Relax numeric variables
ds28	**3**	-	**3**	-	Relax Cat. 1
ds29	**3**	-	**3**	-	Relax Cat. 2

Table 22: Estimated numbers of clusters by four validity indices (continued).

When relaxing some attributes, from the result of *ds26* in Table 22, these methods are not good at handling pure numeric attributes since the *BK* index catches four and other three fail. On the contrary, three indices can better handle pure categorical attributes as seen in *ds27*.

The above two tables show that the *BK* index is more accurate than the *DU* and *SI* indices. The *CH* index is not capable of handling the synthetic datasets because the plot is non-decreasing with respect to *k*, meaning we cannot find a maximum number. Three indices handle perfectly for the well-separated dataset without adding noise. However, the *BK* index outperforms the *DU* and *SI* indices when datasets become noisy. The non-Gaussian noise seems no much effect on the indices. The weakness of the indices is the ability to deal with pure numeric datasets.

4.4.2 Real-world Datasets

We work with the six real datasets described in the preceding chapter. The Iris and Iris-Disc datasets consist of three classes. One type of Iris is linearly separable from the other two, but those two overlap. As a result, the index will be tested to determine whether it can properly deal with overlapping clusters. The DNA-nominal dataset demonstrates a sub-cluster hierarchical structure where three clusters (ie boundary, ei boundary, and no boundary) fall into two pairs (with or without boundary). This dataset is then used to probe whether the index could recognize the sub-cluster hierarchical structure.

In a similar way, we use the six corresponding trees derived in Chapter 3, and calculate the validity indices for the six real datasets, B(*k*), with respect to *k* from two up to 18, and plot the results in Figure 5.

The correct number of the true clusters and the estimated numbers of clusters by the four indices for the six real datasets are provided in Table 23. The results favor the *BK* index out of the four indices. The *BK* index obtains the correct number of clusters for Heart Disease, Vote, Austrian Credit and DNA-nominal datasets; but for Iris and Iris-disc datasets, it ignores the two overlapping clusters and only catches two types of Iris.

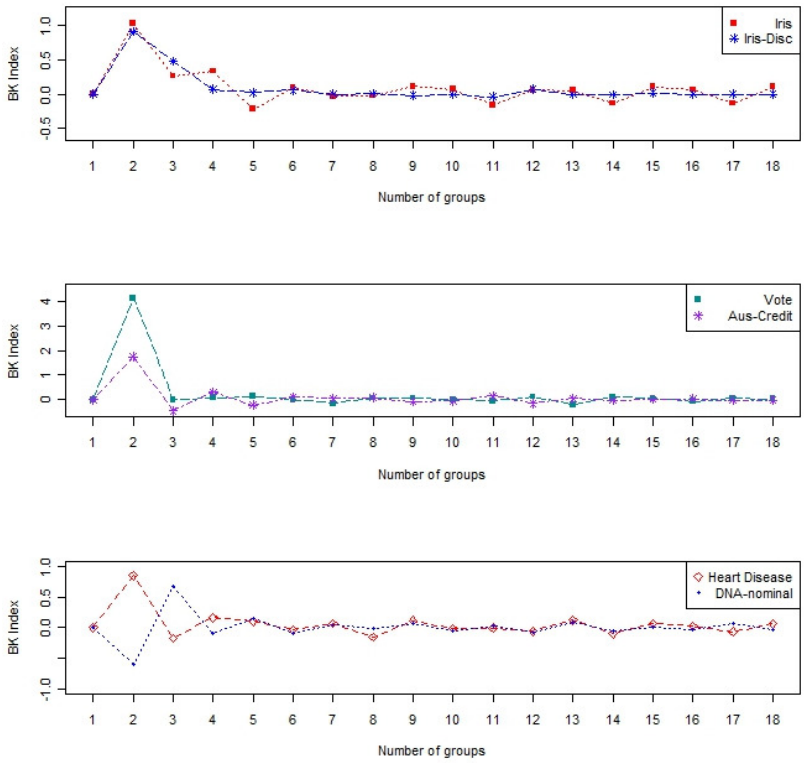

Figure 5: B(*k*) for six real-world datasets.

Dataset	# Cluster (true)	*BK*	*CH*	*DU*	*SI*
Heart Disease	2	**2**	**2**	-	**2**
Iris	3	2	-	2	2
Iris-Disc	3	2	-	-	-
Vote	2	**2**	**2**	**2**	**2**
Aus-Credit	2	**2**	**2**	3	3
DNA-nominal	3	**3**	2	2	2

Table 23: Estimated numbers of clusters by four validity indices for real datasets.

All of the cluster validation methods cannot catch three types of Iris and fail to detect the two overlapping clusters, but obtain the true class number of the Vote dataset. All validity indices except the *DU* index capture the correct group of patients in the Heart Disease dataset. For the Austrian credit dataset, the *BK* and *CH* retrieve the correct number; by contrast, the *DU* and *SI* estimate one more cluster. The DNA dataset has a cluster hierarchy where some clusters are closely grouped together. Only the *BK* index obtains the correct number of DNA dataset, while other three confuse the subclass structure and get two rather than three.

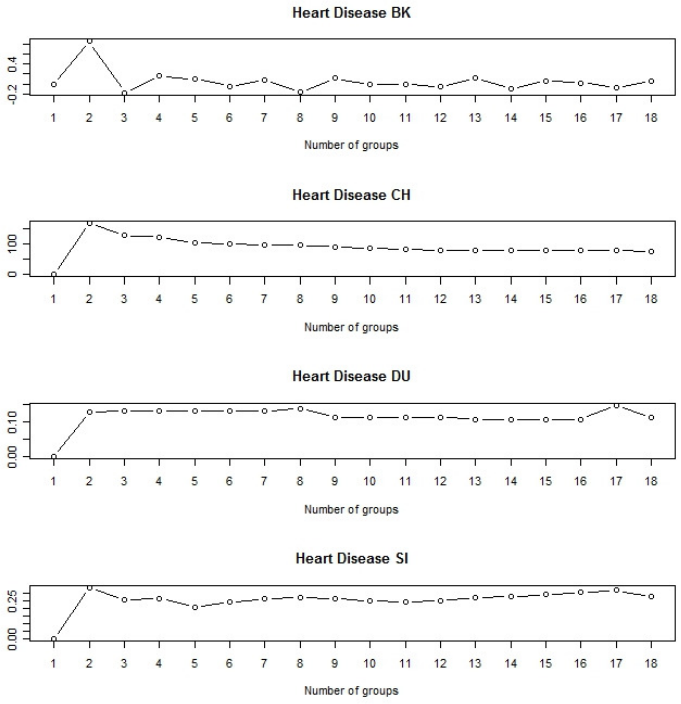

Figure 6: Plots of four indices on Heart Disease.

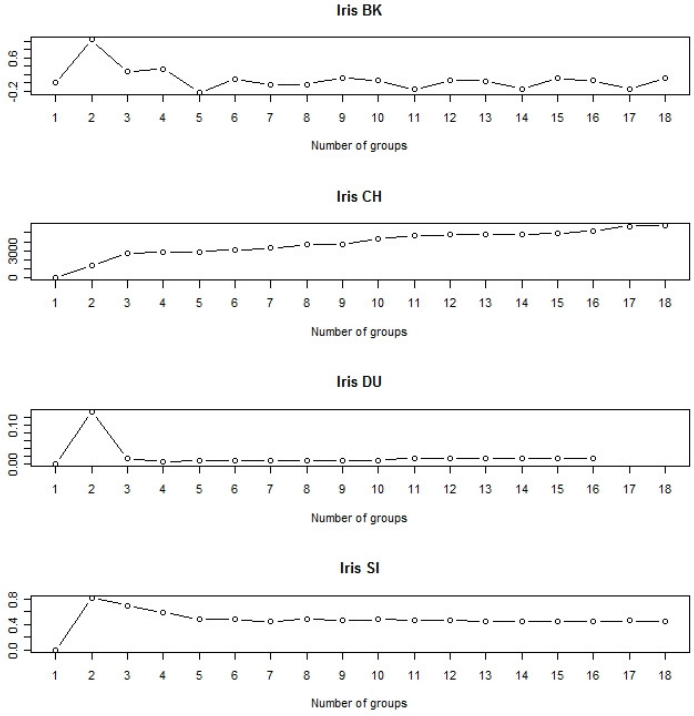

Figure 7: Plots of four indices on Iris.

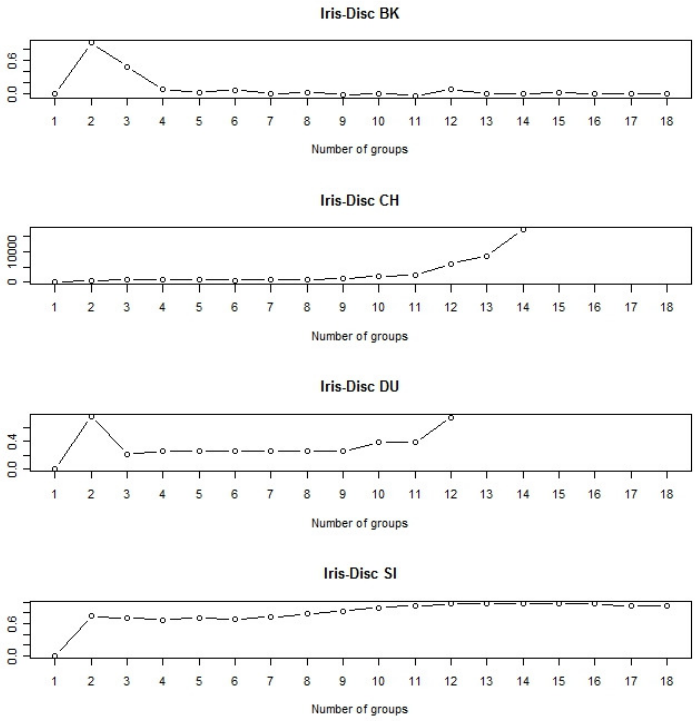

Figure 8: Plots of four indices on Iris-Disc.

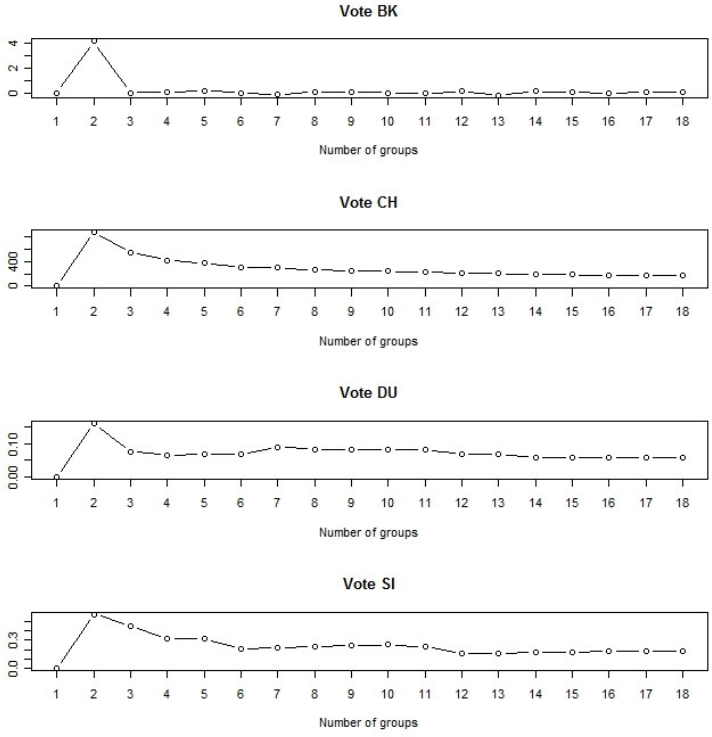

Figure 9: Plots of four indices on Vote.

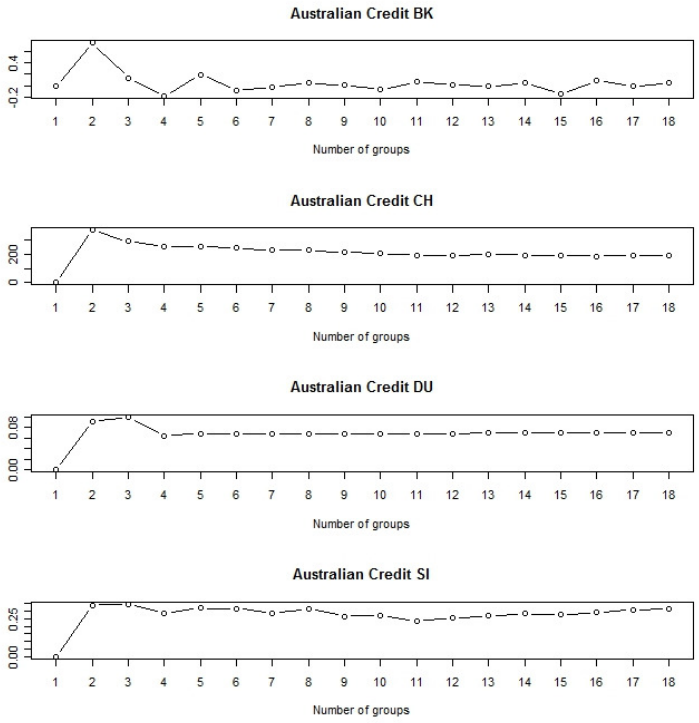

Figure 10: Plots of four indices on Australian Credit.

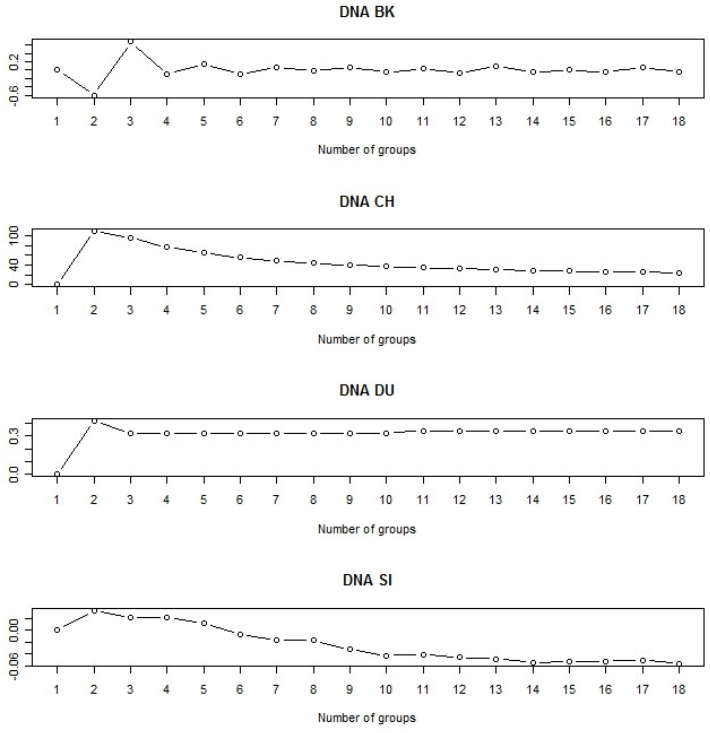

Figure 11: Plots of four indices on DNA.

4.4.3 Preprocessed Real Datasets

In this section, we conduct a comparative study of the four indices on the series of datasets generated by data preprocessing on the six real-world datasets, and especially, focus on the datasets that achieve the most accuracy when using the co-occurrence distance, which are indicated in the second column in the tables (Table 24 – Table 28) in this section. The next four columns show the estimated numbers of clusters by four indices; and the last column presents brief description for each dataset. As usual, -'s represent the failed methods. The bold font denotes the value equal to the true number of classes.

4.4.3.1 Iris Dataset

There are three types of Iris in the Iris datasets. The *BK*, *CH*, and *DU* indices only capture two types in the original dataset. However, for the two datasets of interest, namely, Iris 2 and Iris 5, the *BK* index identifies the number correctly, but all other indices fail.

		BK	***CH***	***DU***	***SI***	**Description**
Iris		2	-	2	2	
Iris 1		2	-	6	7	Discretize SW and SL; PW, PL
Iris 2	√	**3**	-	5	5	Discretize PW and PL; SW and SL
Iris 3		2	-	-	-	Four real and four categorical Attr.
Iris 4		**3**	-	-	6	Discretize SW and SL
Iris 5	√	**3**	-	-	5	Discretize PW and PL

Table 24: Estimated numbers of clusters by four validity indices for Iris.

The plots of four indices present some details. For instance, the *DU* index on Iris 2 shows the hard decision between 2 and 5 while the *SI* index is confused among 3 – 6.

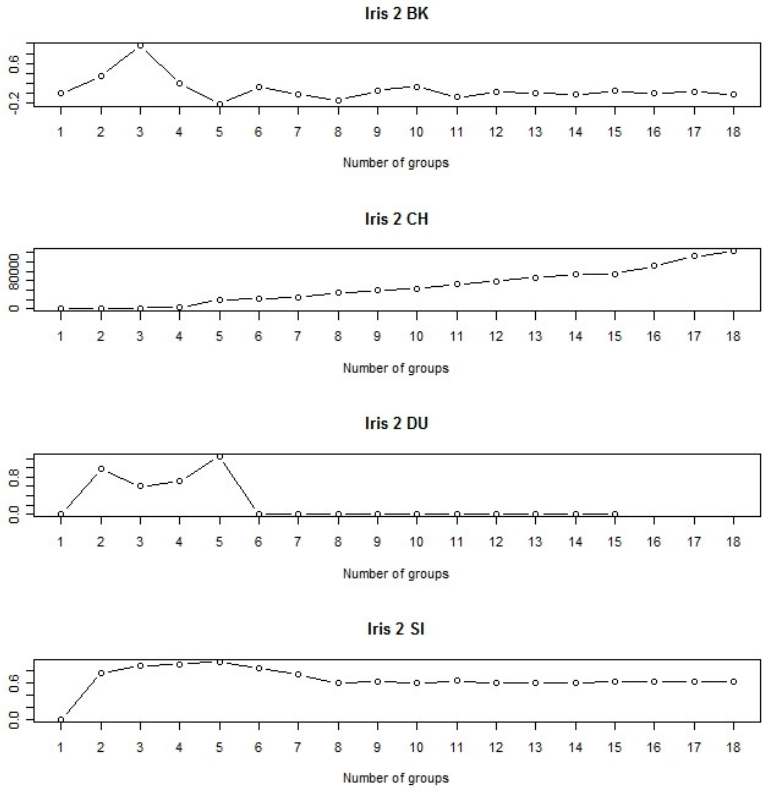

Figure 12: Plots of four indices on Iris 2.

4.4.3.2 Vote Dataset

All numbers of clusters for these vote datasets caught by the *BK* index are correct. However, from Table 25, other three indices almost fail to find the proper number.

		BK	***CH***	***DU***	***SI***	**Description**
Vote		**2**	**-**	**2**	**2**	
Vote 1	√	**2**	-	-	-	Significant Attr. (C3, C4)
Vote 2	√	**2**	-	-	-	Significant Attr. (C4, C8)
Vote 3	√	**2**	-	7	-	Significant Attr. (C3, C4, C5)
Vote 4		**2**	-	-	-	Significant Attr. (C3, C4, C5, C8)
Vote 5		**2**	**2**	-	**2**	The nine weakest Attr.
Vote 6		**2**	-	3	-	The four Attr. among the weakest

Table 25: Estimated numbers of clusters by four validity indices for Vote.

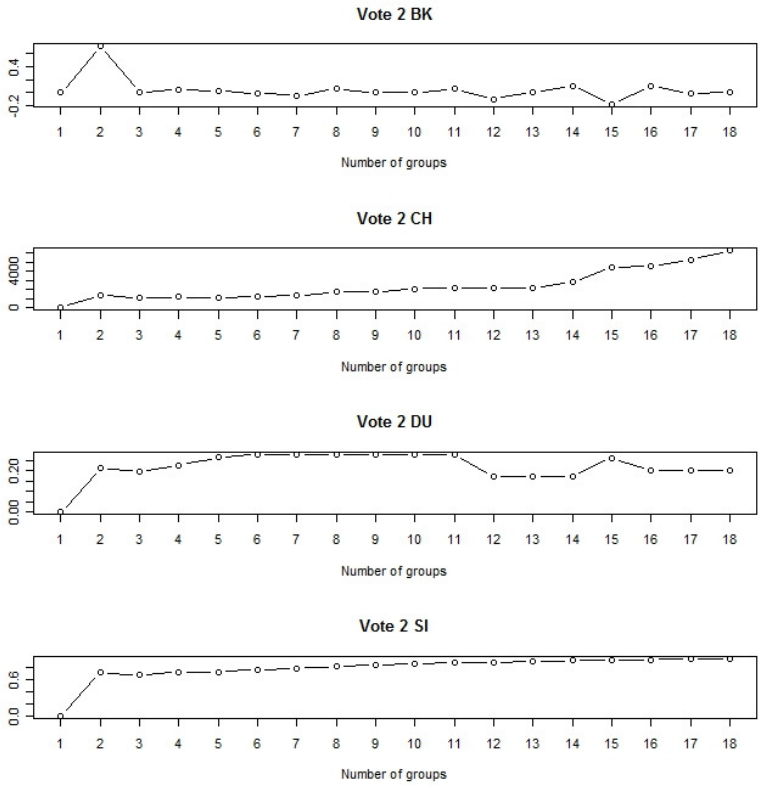

Figure 13: Plots of four indices on Vote 2.

4.4.3.3 Heart Disease Dataset

Only the *BK* index captures the correct number of clusters for all constructed datasets based on the Heart Disease dataset. By contract, from Table 26, other three indices almost fail to find the proper number.

		BK	***CH***	***DU***	***SI***	**Description**
Heart		**2**	**2**	-	**2**	
Heart 1	√	**2**	3	-	-	Significant Attr. (C4, C8, R3, R5)
Heart 2	√	**2**	-	6	7	Significant Attr. (C4, C8, R3, R4, R5)
Heart 3		**2**	-	6	7	C2, C4, C5, C8, R1, R2, R3, R4, R5
Heart 4		**2**	**2**	3	-	All categorical attributes
Heart 5	√	**2**	-	-	**2**	All numeric attributes

Table 26: Estimated numbers by four validity indices for Heart Disease.

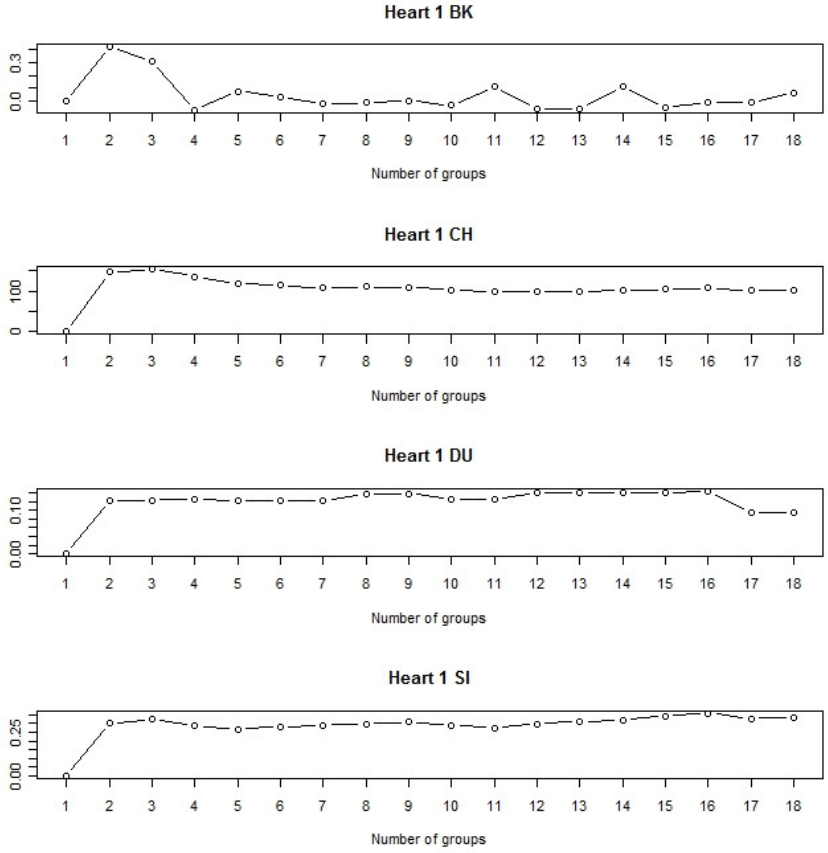

Figure 14: Plots of four indices on Heart 1.

4.4.3.4 Australian Credit Dataset

The *BK* and *CH* indices have good performance on datasets related to the Australian Credit data. They detect the correct number of clusters for all datasets except Aus 5 dataset, which is very noisy since the highest clustering accurate rate in Aus 5 is 68.12% compared to 84.78% in original dataset (Aus). The corresponding accuracy can be found in Section 3.5.4.

		BK	***CH***	***DU***	***SI***	**Description**
Aus	√	**2**	**2**	3	3	
Aus 1		**2**	**2**	3	3	C2, C4, C5, C6, C8, R3, R4
Aus 2	√	**2**	**2**	**2**	-	C2, C3, C4, C5, C6, C8, R3, R4
Aus 3	√	**2**	**2**	**2**	-	C2, C3, C4, C5, C6, C8
Aus 4	√	**2**	**2**	5	-	All categorical attributes
Aus 5		6	-	-	**2**	All real attributes

Table 27: Estimated numbers by four validity indices for Australian Credit.

4.4.3.5 DNA-nominal Dataset

The *BK* index is able to catch the correct number of datasets related to the DNA data. On the other hand, the *CH*, *DU*, and *SI* indices find two subclasses of DNA in all datasets except DNA 3. However, as could be seen in Figure 15, the estimated numbers of the other three indices for DNA 3 are not stable and easily confused since the values on 2 and 3 are very close in the three plots. If some redundant information or noise is introduced, for instance, adding attribute 1, 53, 55 and 57, the three indices are not able to determine the proper number and catch two rather than three.

		BK	***CH***	***DU***	***SI***	**Description**
DNA	√	**3**	2	2	2	
DNA 1		**3**	2	2	2	First 12 most significant Attr.
DNA 2	√	**3**	2	2	2	First 38 most significant Attr.
DNA 3	√	**3**	2	**3**	**3**	All Attr. except C1,C53,C55,C57

Table 28: Estimated numbers of clusters by four validity indices for DNA.

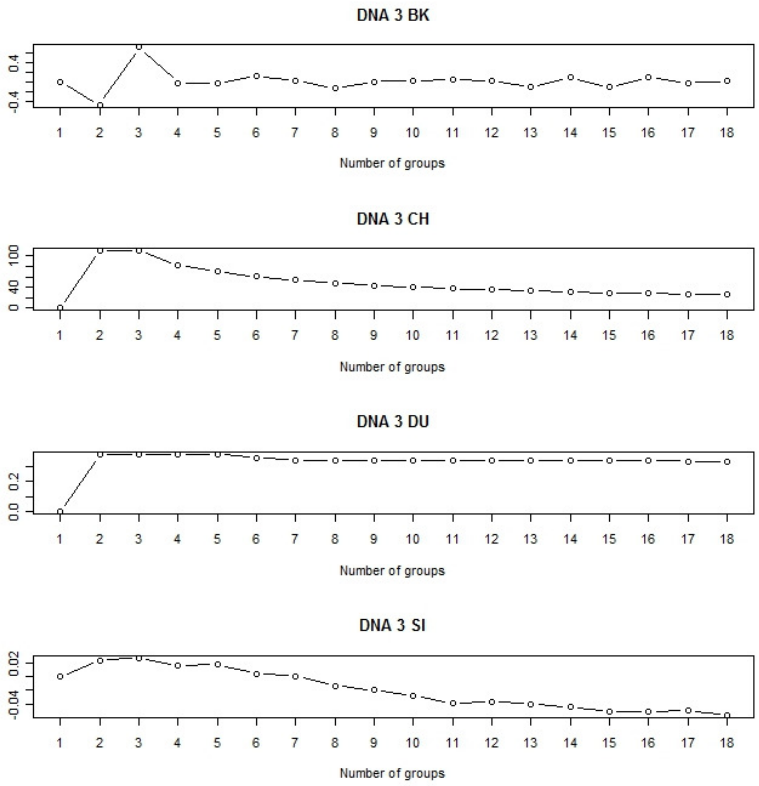

Figure 15: Plots of four indices on DNA 3.

4.5 Summary

The proposed algorithm is performed on some synthetic and real-world datasets with various characteristics. The results show it is efficient not only to cluster a dataset having mixed types of features, but also to determinate the best number of classes. The *BK* index presents an impressive result in comparison with the *DU*, *CH*, and *SI* indices.

Especially, we provide the solution to preprocess the mixed data according to the ranks of the importance for each attribute and properties of hierarchical clustering with the co-occurrence distance. As a result, the reduced mixed datasets are more applicable to be analyzed by the proposed algorithms in terms of not only the capability of achieving a higher accurate rate, but also the ability to find the best number of groups.

CHAPTER 5 CONCLUSION

Many applications give rise to databases with mixed data, that is, variables that take both numerical and categorical attributes. As one example, a surveillance database of criminal activities might contain numeric attributes such as age, time of the day, and number of the offenders, as well as categorical attributes like gender, location, and weapons used (Yang and Olafsson, 2011). It is often of interest to find natural clusters of instances in such databases, but unfortunately the majority of clustering algorithms are designed for only one data type and incapable of handling data containing both types directly.

Motivated by the need to solve mixed data type clustering problems and the current gap in the literature regarding methods for such problems, in this dissertation we propose and demonstrate a clustering framework that is effective and yields important practical results. This framework has two main components. First there is the actual clustering algorithm, which is based on traditional hierarchical clustering and outputs a tree structure containing multiple actual cluster solutions. For measuring similarly, we choose the recently proposed co-occurrence measure. We compare this measure with three other well-known distances measurements capable of handing mixed data when incorporated into agglomerative hierarchical clustering. These measures are the Goodall distance, the *k*-prototype distance, and the optimal weight distance. We also identify certain limitations of applying hierarchical clustering with a co-occurrence distance and propose a solution in which the co-occurrence distance would outperform other distance measures.

The second component of the framework is to define a validity index to find an optimal number of clusters in mixed datasets and integrate it with the hierarchical clustering. The performance of the so-called *BK* index is compared to other known validity indexes, namely the Calinski-Harabasz index (*CH*), the Dunn index (*DU*), and the Silhouette index (*SI*), and the results are favorable for using the *BK* index for cluster validation in mixed data.

By testing the proposed approach on both standard benchmark datasets from the UCI repository and, synthetic datasets with various characteristics, we demonstrate the method not only effectively retrieves the true class in terms of prediction accuracy, but also is capable of effectively finding the true number of clusters.

In conclusion, our framework addresses two important issues regarding clustering mixed datasets. One is how to search for the optimal number of clusters, which is important as this is unknown in many applications. We extend the *BK* index to both data types. Thus, it would be used to quantify clustering results from the hierarchical algorithm. The *BK* index outperforms other three indices, namely, the *CH*, *DU*, and *SI* indices in comparison with the true numbers of clusters. The other issue is how to group the objects "naturally" given the number of cluster. We use the co-occurrence distance to measure the dissimilarity since this distance is as effective as other distances capable of handling mixed data such as the Goodall, *k*-prototype and optimal weighted distances.

All of the research problems considered in this dissertation address critical issues for clustering mixed-type attributes in data mining applications. Clearly the research in this area is far from complete. Some details would be improved such as using optimal techniques to discretize the numeric attribute rather than the five-equal width method. Feature selection in the proposed algorithm is optional and exploratory, but was found to be promising. However, providing an adaptive feature selection technique to systemically and dynamically determine which attributes should be included as a preprocessing step prior applying learning algorithms is also another challenge. In addition, as instances accumulate, scalability improvement will be under consideration, which leads to solving optimization problems on instance selection. On the other hand, feature selection and instance selection may provide valuable information about the objects of interest.

BIBILIOGRAPHY

Ahmad, A. and Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering, 63*, 503–527.

Bezdek, J. and Pal, N. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics, 28(3),* 301–315.

Biswas, G., Weingberg, J., Fisher, D.H. (1998). ITERATE: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man, and Cybernetics*, *28*, 219–230.

Brouwer, R., Groenwold, A. (2010). Modified fuzzy c-means for ordinal valued attributes with particle swarm for optimization. *Fuzzy Sets and Systems, 161,* 1774−1789.

Cai, Z., Wang, D., Jiang L. (2007). *K*-distributions: A new algorithm for clustering categorical data. In: *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE, 4682,* 436–443.

Celeux, G. and Govaert, G. (1991). Clustering criteria for discrete data and latent class models. *Journal of Classification, 8*, 157–176.

Cheesman, P. and Stutz, J. (1995). Bayesian classification (AUTO-CLASS): Theory and results. In: *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge.

Chen, K. and Liu, L. (2009). "Best K": critical clustering structures in categorical Datasets. *Knowledge Information System, 20,* 1–33.

Chen, N., Marques, N. (2005). An extension of self-organizing maps to categorical data. In: *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE, 3808,* 304–313.

Davies, D. and Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Patterns Analysis and Machine Intelligence, 1,* 224−227.

Day, W. and Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, *1*, 7–24.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, and V. (2004). Clustering large graphs via the Singular Value Decomposition. *Machine Learning*, *56*, 9–33.

Dunn, J.C. (1974). Well separated clusters and optimal fuzzy partitions. *Journal of cybernetics, 4,* 95−104.

Fisher, D.H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2 (2)*, 139–172.

Fowlkes, E.B. and Mallows, C.L. (1983). A method for comparing two hierarchical clusterings. *J. Amer. Statist. Assoc., 78 (383),* 553–569.

Ganti, V., Gehrke, J., Ramakrishnan R. (1999). Cactus—clustering categorical data using summaries. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 73–84.

Goodall, D. W. (1966). A new similarity index based on probability. *Biometrics, 22*, 882−907.

Gowda, K.C., and Diday, E. (1991). Symbolic clustering using a new dissimilarity measure. *Pattern Recognition, 24,* 567–578.

Guha, S., Rastogi, R., Kyuseok S. (1999). ROCK: A robust clustering algorithm for categorical attributes. In: *Proceedings of 15th International Conference on Data Engineering*, Sydney, Australia, 512–521.

Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2002). Cluster validity methods: Part I and II. *SIGMOD Rec*, *31(2),* 40–45.

Huang, Z. (1998). Extensions to the K-modes algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2 (3),* 283–304.

Huang, Z., and Ng, M.K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems, 7 (4),* 446–452.

Hsu, C., Chen, C., and Su, Y. (2007). Hierarchical clustering of mixed data based on distance hierarchy. *Information Sciences,177*, 4474–4492.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data.* New Jersey: Prentice Hall.

Jiau, H., Su, Y., Lin, Y., Tsai, S. (2006). MPM: a hierarchical clustering algorithm using matrix partitioning method for non-numeric data. *Journal of Intelligent Information Systems, 26,* 185–207.

Jing, L., Ng, M., and Huang, J. (2007). An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, *IEEE Transactions on Knowledge and Data Engineering, 19,* 1026−1041.

Jing, R., Zliao, X., Yu, J. (2007). A theoretical analysis framework of customer cluster value of dynamic B2C E-Commerce. In*: Proceedings of the 3rd International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China, 3438–3441.

Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* New York, Wiley.

Kim, K., Chung, H., Rha, S. (2009). A weighted sample size for microarray datasets that considers the variability of variance and multiplicity. *Journal of Bioscience and Bioengineering, 108,* 252–258.

Kim, M. and Ramakrishna, R. (2005). New indices for cluster validity assessment. *Pattern Recognition Letters, 26(15)*, 2353−2363.

Lam, Y. and Yan, H. (2005). A new cluster validity index for data with merged clusters and different densities. In *2005 IEEE International Conference on Systems, Man and Cybernetics, V1,* 798−803.

Lee, M. (2009). On Fuzzy Cluster Validity Indices for the Objects of Mixed Features. In *Proceedings of the 18th international conference on Fuzzy Systems,* 390−395.

Lee, M., and Pedrycz, W. (2009). The fuzzy C-means algorithm with fuzzy P-mode prototypes for clustering objects having mixed features. *Fuzzy Sets and Systems*, *160,* 3590–3600.

Li, C., and Biswas, G. (2002). Unsupervised Learning with Mixed Numeric and Nominal Data. *IEEE Transactions on Knowledge and Data Engineering, 14(4),* 673−690.

Liao, S., Ho, H., Lin, H. (2008). Mining stock category association and cluster on Taiwan stock market. *Expert Systems with Applications, 35,* 19–29.

Mateo, R., Salvo, M. et al. (2008). Balanced Clustering using Mobile Agents for the Ubiquitous Healthcare Systems. In*: Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology*, Busan, South Korea, 686–691.

Modha, D. and Spangler W. (2003). Feature Weighting in *k*-Means Clustering. *Machine Learning, 52,* 217–237.

McKusick, K., Thomson, K. (1990). COBWEB/3: A portable implementation, Technical Report FIA-90-6-18-2, NASA Ames Research Center.

Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. *J. Amer. Statist. Assoc., 66,* 846–850.

Reich, Y., Fenves, S.J. (1991). The formation and use of abstract concepts in design. In: D.H. Fisher, M.J. Pazzani, P. Langley (Eds.), *Concept Formation: Knowledge and Experience in Unsupervised Learning*, Morgan Kaufman, Los Altos, Calif, 323–352.

Suikki, L., Takala, J., Vauhkonen, T. (2006). Development of the competitiveness of a manufacturing network by using cluster theory - A case study in aluminium network of alajarvi. In: *Proceedings of the 8th International Conference on Industrial Logistics*, Kaunas, Lithuania, 260–271.

Touray, K., Adetifa, I., Jallow, A., et al. (2010). Spatial analysis of tuberculosis in an Urban West African setting: is there evidence of clustering? *Tropical Medicine & International Health, 15,* 664–672.

Yang, M., Hwang, P., Chen, D. (2004). Fuzzy clustering algorithms for mixed feature variables. *Fuzzy Sets and Systems, 141,* 301–317.

Yang, R., Olafsson, S. (2011). Classification for predicting offender affiliation with murder victims. *Expert Systems with Application, 38(11),* 13518–13526.

Yasunori, E., Yukihiro, H., Sadaaki, M, (2007). Agglomerative hierarchical clustering for data with tolerance. In: *Proceedings of 2007 IEEE International Conference on granular computing*, CA, USA, 404–409.